

Morichal Software

Juan Vargas

rootweiller83@gmail.com

Morichal Software

by Juan Vargas

Published \$Date: 2017/07/07 10:27:39 \$

Copyright © 2017 Juan Vargas

Hace un tiempo atrás me propuse la idea de crear un software que permitiera a docentes, alumnos, padres y representantes tener un control total sobre las actividades que suceden dentro de una institución académica, en el sentido, de poder controlar notas, asignaturas, contenidos programáticos. Para el docente y la sociedad actual es esencial manejar estadísticas de todo tipo de cosas, en la educación (por lo menos en Venezuela) se manejan muy pocas, algunas están desfasadas con el tiempo ya, otras son de carácter privado y el fin acá, es que puedan contar con una API donde se conecten y evalúen, por ejemplo, cual ha sido el promedio de notas en una asignatura de un municipio en un estado X, comparar ese promedio con otro municipio en otro estado y a nivel social, poder determinar por que en un u otro los alumnos tienen mejor o peor promedio.

Morichal, nació hace un tiempo, mientras trabajaba como docente de aula, siempre quise conocer los números que rodeaban la educación, promedios generales, asignaturas, docentes mejor evaluados y esas cosas, por ende, decidí crear Morichal, al principio era solo para el liceo donde trabajaba, sin embargo había un enfoque mas de llevar el control del departamento de evaluación, luego deje ese trabajo y pues coloque a un lado el proyecto, sin embargo lo ando retomando, pues veo la necesidad de mejorar la calidad educativa en muchos aspectos, para poder comenzar con esas mejoras es necesario saber donde se esta y hacia donde se va.

Table of Contents

1. Planteamiento del Problema	1
1.1. Planteamiento del Problema.....	1
2. Requisitos Funcionales	2
2.1. Requisitos Funcionales	2
3. Modelo de Negocio	4
3.1. Modelo de Negocio	4
4. Requerimientos de Software	5
4.1. Control del Documento	5
5. Reglas de Confidencialidad	6
5.1. Reglas de Confidencialidad.....	6
6. Referencias	7
6.1. referencias	7
7. Arquitectura de Software.....	8
7.1. Descripción	8
8. Contextualización del Proyecto	10
8.1. Descripción	10
8.2. Resumen Conceptual del Proyecto.....	10
9. Especificación de Casos de Uso.....	11
9.1. Introducción	11
10. Instalación del Entorno de Desarrollo	12
10.1. Preparación del Sistema	12
10.2. Configuraciones Nginx	13
11. Conclusión	14
11.1. Conclusión	14

List of Tables

5-1. Referencias	6
10-1. Paquetes	12
10-2. Servidor Backend	12

Chapter 1. Planteamiento del Problema

1.1. Planteamiento del Problema

Desde el inicio de la informática el software de computadoras era libre, los programadores se intercambiaban piezas de software sin ningún problema, siempre que se respetara la propiedad intelectual, sin embargo, las cosas comenzaron a cambiar cuando el computador se hizo más accesible a todas las personas, comenzaron a aparecer empresas que desarrollaban software para computadoras y estas comenzaron a ver el mismo como una pieza del negocio, su rentabilidad se basó en la venta de copias, pero que no te pertenecían sino más bien adquirirías un permiso de uso de parte de la empresa, el conocimiento de como estaba hecho el software lo encerraron solo para esa empresa. El tiempo fue pasando y en una carrera contra estas empresas un programador decide romper con los paradigmas impuestos por las casas de software y comienza a liberar su software, así como también escribe la primera versión de la Licencia Pública General GPL con la cual pretendía devolverle a los programadores la libertad de poder compartir su software entre ellos y otras personas sin ser acusados de piratas.

El software privativo continua creciendo ya que no es solo el desarrollo de una aplicación que ejecute un proceso y que la empresa cierre ese código, también se basa en crear patentes para así generar un monopolio en donde solo la empresa que desarrolla esa solución tecnológica sea la que tiene el poder de aprovecharla y ganar dinero con ella. El software libre es, más allá del tema filosófico y técnico, una solución real a muchos de los problemas de la tecnología, con él es posible, compartir no solo el conocimiento técnico, también es posible unir esfuerzos globales para solucionar problemas que pudieran afectar a muchos y colocar esa solución accesible a todos, para que todos se beneficien de ello, no se dice que con software libre no se pueda ganar dinero o que eso este mal visto, al contrario, se puede y se debe ganar dinero con software libre, la principal diferencia es que la solución queda accesible y el código fuente puede ser tomado por cualquier persona en el mundo, adaptarlo a su entorno y resolver su problema.

Desde hace varios años en Venezuela se utiliza un software mediante el cual se lleva la automatización del departamento de control de estudios y evaluación de una institución pública, el cual está basado en software privativo, el mismo genera una rentabilidad anual por el pago de licencias de uso y no tienen acceso al código fuente, esto genera una violación a la Ley de Infogobierno la cual establece en su artículo 34: "Artículo 34. El desarrollo, adquisición, implementación y uso de las tecnologías de información por el Poder Público, tiene como base el conocimiento libre. En las actuaciones que se realicen con el uso de las tecnologías de información, sólo empleará programas informáticos en software libre y estándares abiertos para garantizar al Poder Público el control sobre las tecnologías de información empleadas y el acceso de las personas a los servicios prestados. Los programas informáticos que se empleen para la gestión de los servicios públicos prestados por el Poder Popular, a través de las tecnologías de información, deben ser en software libre y con estándares abiertos".

Chapter 2. Requisitos Funcionales

2.1. Requisitos Funcionales

Los requisitos de un sistema son aquellos requerimientos propios del sistema a desarrollar y comprenden una serie de funciones que el sistema debe realizar. A continuación se definirán los requisitos funcionales y no funcionales del sistema "Morichal".

- Requisitos Funcionales

- El sistema enviará un correo electrónico cuando se registre una de las siguientes acciones: registro de personal docente, administrativo, registro de alumnos, envío de boletines.
- Se permitirá el registro de alumnos con la siguiente información incompleta: correo electrónico, número de teléfono, los cuales podrán ser completados posteriormente.
- El sistema permitirá visualizar sin permiso la data estadística de: cantidad de alumnos registrados, cantidad de instituciones
- El sistema permitirá la carga por parte del docente de las notas por alumno y por sección
- Se permitirá la creación de los siguientes perfiles: Institución educativa, secciones, docentes
- A cada perfil se le asignará un identificador automático, pero la búsqueda de perfiles se hará mediante el documento de identidad en el caso de los docentes y el RIF o razón social en el caso de las instituciones.
- El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña.
- El software estará basado en software como servicio (SaaS), pudiéndose ejecutar en sistemas operativos: Windows, GNU/Linux, IOS, Android, Windows Phone.
- El software podrá usarse sin software adicional en el dispositivo en el que se ejecute.

- Requisitos no Funcionales

- El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos
- Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema
- La aplicación web debe poseer un diseño "Responsive" a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes
- El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final

- La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema
- La interfaz de usuario será implementada para navegadores web únicamente con HTML5 y JavaScript
- El sistema debe asegurar que los datos estén protegidos del acceso no autorizado

Chapter 3. Modelo de Negocio

3.1. Modelo de Negocio

El modelado de negocios se describe como la manera de implementar y hacer que este software sea sustentable y sostenible para ello nos basaremos en la filosofía del software libre, la cual reza que aunque no se permite la venta de licencias de uso, si es posible generar ingresos a través de soporte, mantenimiento e instalación del software, en consecuencia el modelo se basa en el soporte que se le brindará a las diferentes instituciones.

Es de destacar que todo el código fuente, así como la documentación estará disponible a través de herramientas tecnológicas que permitan la difusión total del conocimiento y que no existirá software oculto ni aplicaciones de terceros privativas en "Morichal", las colaboraciones de terceros serán aceptadas a través del sistema tickets de control de versiones que se escoja para la publicación del software, así como las discusiones que surjan acerca de mejoras e implementaciones nuevas se harán a través del sistema de listas de correos dispuestos y público que tendrá el proyecto.

Teniendo una cuenta que "Morichal" está basado en software libre, programadores, pequeñas y medianas empresas del sector tecnológico, podrán hacer uso del software colocando sus plataformas tecnológicas y vendiendo el hosting, mantenimiento y configuración del mismo a través de la arquitectura "Software as Service" (Software como Servicio, SaaS), esta no es la vía más práctica para algunas instituciones, pero sirve para aquellas que poseen pocos equipos de computación en sus instalaciones.

Chapter 4. Requerimientos de Software

4.1. Control del Documento

Especificación de requerimientos de software para Morichal. Versión 1.0

Chapter 5. Reglas de Confidencialidad

5.1. Reglas de Confidencialidad

El presente documento está abierto y publicado bajo la licencia GPL, tanto el software desarrollado como el fuente del documento estarán publicados en el repositorio de github del proyecto.

Table 5-1. Referencias

Documento del Sistema	Arquitectura Tecnológica	Especificación de casos de uso
Publicado en github en el repositorio de Documentación	Presentada en imagenes dentro del documento del sistema	Presentado dentro del documento del sistema

Chapter 6. Referencias

6.1. referencias

El objetivo de esta sección es especificar los diferentes requerimientos del software para "Morichal", dentro de los mismos estarán definidos los módulos de estudiantes, profesores, salones de clases, asignaturas y colegios.

Chapter 7. Arquitectura de Software

7.1. Descripción

Los sistemas cliente-servidor están desarrollados de tal manera que los clientes sean capaces de realizar una consulta al servidor y este les responda, aunque la idea inicial es que todo se encuentre entre un grupo de computadoras es también válido para que este modelo sea ejecutado desde un mismo equipo de computación. Lo que se busca con ello es la posibilidad de distribuir los procesos y acciones del sistema entre varios equipos con la finalidad de mejorar los tiempos de respuesta, la seguridad de los datos y la disponibilidad de datos en todo momento.

Dentro del esquema planteado para el desarrollo de "Morichal", se encuentra la arquitectura cliente-servidor, en el caso de algunas instituciones les servirá para hacer accesible la información bien sea, dentro de la red local o en una red externa (conectado a Internet), así mismo el desarrollo de "Morichal" permite la posibilidad de usar una arquitectura un tanto más avanzada, como la de software como servicio (SaaS), la cual consiste en que el código fuente sea tomado por una empresa y/o programador, para ser colocado dentro de su infraestructura tecnológica y cobrar el soporte, mantenimiento y respaldo de data de esa institución. Todo ello es posible gracias a la manera en la que se desarrolla "Morichal".

"Morichal" está basado en una arquitectura de Cliente-Servidor, con ella se busca agilizar las tareas y crear un sistema distribuido por un conjunto de servidores que son capaces de responder a las diferentes peticiones de los clientes, así mismo su estructura de desarrollo se basa en una aplicación web, la cual es accesible desde cualquier navegador web moderno, permitiendo al usuario escoger el sistema operativo del lado cliente.

En la imagen 1, se describe el como funcionaria en una arquitectura cliente-servidor distribuida en el centro se especifican 3 discos en los cuales se distribuye la base de datos, el backend desarrollado con Python y el frontend desarrollado con HTML5, alrededor se visualizan los clientes que se conectan a través del protocolo de comunicaciones HTTP utilizando como puente el navegador.

Como se presenta en la tabla 1, el sistema va a permitir a un administrador general de la plataforma el registro tanto de docentes como alumnos en la misma, a su vez de complementar la información necesaria de la institución educativa, esto ayudará a que los reportes de la aplicación sean mucho más completos. Los datos en la plataforma estarán resguardados mediante tablas en el sistema, en una base de datos basada en el motor PostgreSQL, pudiendo ser ejecutada en otros motores como lo son (MariaDB, MySQL, SQLite), en la siguiente gráfica se muestra la relación que existe entre cada una de las tablas del sistema.

Cada perfil de usuario, entendiendo como perfil a un docente, alumno, padre, madre y/o representante, es modelado en tablas con características especiales para diferenciarlos, es decir, existe una tabla principal que guardalos datos básicos como lo son nombre, apellido, usuario, password, id, si es usuario activo, si

es un usuario administrador (en el caso de los docentes)

En la imagen anterior se muestra como se distribuye la tabla teachers, existe una relación de esa tabla con la tabla user que como ya se indicó se estructura de la siguiente manera.

A lo largo del desarrollo de los módulos de la plataforma se hicieron diferentes pruebas unitarias al mismo, también pruebas de rendimiento de la aplicación. En las diferentes pruebas que se hicieron se pudo constatar que efectivamente la aplicación cumple con los requerimientos iniciales de su desarrollo, veamos algunas pantallas que así lo demuestran.

Está es la primera pantalla de la aplicación, está desarrollada para no permitir el acceso a usuarios no registrados, los usuarios deben ser creados previamente por el administrador del centro educativo. Una vez se logra acceder al sistema, se continua con el dashboard el cual en esta primera versión tiene el siguiente diseño

El dashboard es la pantalla administrativa de la aplicación en ella se puede apreciar el diseño inicial de la aplicación, esta pantalla que semuestra forma parte de un perfil administrativo el cual tiene permisos de visualizar todos los alumnos, profesores, representantes y los mensajes generados por la plataforma, así mismo tiene el acceso a gráficos estadísticos que le indican los diferentes rendimientos de los alumnos por asignatura y/o por salón de clase.

- Books
 - Donald E. Knuth - The Art Of Computer Programming
 - Nils J. Nilsson - Artificial Intelligence: A New Synthesis
 - Pure Mathematics 2 - Geoff Mannall, Michael Kenwood
- Games
 - Chess
 - Backgammon
 - Noughs And Crosses

Chapter 8. Contextualización del Proyecto

8.1. Descripción

En esta sección se realizará un resumen sustentable de la organización del equipo de trabajo para el proyecto "Morichal", permitiendo definir el alcance del mismo, artefactos y otros aspectos relacionados.

8.2. Resumen Conceptual del Proyecto

En resumidas cuentas "Morichal", comprende una serie de herramientas que le permiten a docentes, alumnos, personal administrativo y padres, establecer una comunicación directa y efectiva, usando tecnologías de información, con el cual podrán acceder a la información en tiempo real de la actuación de sus representados en las diferentes actividades enmarcadas en el calendario académico.

Como ya se explicado "Morichal", nace por la necesidad de conocer en tiempo real las diferentes actividades de los alumnos en el aspecto académico así como generar estadísticas que le permitan a docentes y personal administrativo de las instituciones mejorar aspectos relacionados con el proceso de enseñanza - aprendizaje de los jóvenes.

Entre las cosas a mejorar que deben tener en cuenta los docentes, son la metodología que se usa para la aplicación que prueban el conocimiento, con "Morichal", se busca una evaluación continua de estos aspectos y en base a los resultados de la clase aplicar de manera inmediata los diferentes correctivos a los objetivos planteados, bien sea por parte del docente o por parte de clase. La educación es un proceso de nunca acabar, cada día se debe aprender y aplicar los conocimientos obtenidos en la mejora sustancial de la calidad de vida, de eso se trata, pero mientras no existan indicadores de cuales son las cosas que se deben mejorar, se vuelve un proceso complejo de evaluar.

Chapter 9. Especificación de Casos de Uso

9.1. Introducción

Esta sección describe los casos de uso de los diferentes módulos pertenecientes al software "Morichal", dentro de los cuales se especificarán todos los posibles aspectos de los mismos.

Chapter 10. Instalación del Entorno de Desarrollo

10.1. Preparación del Sistema

El software "Morichal" está desarrollado usando el framework web Django en su versión 1.11.5, así mismo tiene una serie de paquetes los cuales se describen en la siguiente tabla.

Table 10-1. Paquetes

Nombre del Paquete	Versión
Django	1.11.5
virtualenv	15.0.1
Django-Rest-Framework	0.1.0
Psycopg2	2.7.3.1
oauthlib	2.0.4
django-cors-middleware	1.3.1
Pillow	4.2.1
requests	2.18.4

Una de las cosas a tener en cuenta para la puesta en producción o testeo de la aplicación es que puedes hacerlo usando el servidor que trae Django por defecto para estos fines, sin embargo, es de aclarar que esto no es lo recomendable cuando de ponerlo en producción se trata. También puedes usar nginx, apache2 o cualquier servidor web que soporte tecnología wsgi.

El proceso de instalación es bastante sencillo dependiendo de que sistema operativo se vaya a utilizar, para el caso de este documento se va a utilizar la distribución Ubuntu GNU/Linux en su versión 17.10, lo primero que se debe hacer es correr la instalación del paquete `apt install virtualenvwrapper` una vez ejecutada la anterior línea de comando es necesario crear el entorno virtual con el siguiente comando `mkvirtualenv -p /usr/bin/python3 morichal` con el entorno virtual es necesario clonarse el proyecto desde Github `git clone https://github.com/rootweiller/morichal.git` se creará la carpeta del proyecto y dentro de ella está el archivo `pip-requirements.txt`, para instalar las diferentes dependencias del proyecto es necesario ejecutar `pip install pip-requirements.txt`

La aplicación está desarrollada para ser usada en un entorno cliente-servidor, por ello como recomendación se pide que se dividan el servidor del equipo cliente, para el equipo servidor se propone la siguientes características.

Table 10-2. Servidor Backend

Características	Versión
Debian GNU/Linux	9.0
nginx	1.13.5
Disco Duro	60 GB
Memoria RAM	4 GB
Procesador	Intel Xeon o Superior

Para esta versión del proyecto, el backend y el frontend no están divididas, se usa el manejador de templates de Django, sin embargo, la carga de datos se hace a través del admin provisto por Django 1.11.5 y se generaron las API para que se conecten las diferentes versiones de la aplicación, en una primera versión se usará una app híbrida, con la cual se le permitirá a docentes, padres y alumnos cargar la información que se requiera para alimentar de datos al software.

10.2. Configuraciones Nginx

Las configuraciones para levantar el servidor usando nginx 1.13.5 es la siguiente. `server { listen 9000; server_name localhost; access_log /var/log/myproject/myproject_access_log; error_log /var/log/myproject/myproject_error_log; root /home/usuario/dev/mysite/myproject; location /static { alias /home/usuario/dev/mysite/myproject/static; } location / { uwsgi_pass unix:/tmp/uwsgi_myproject.sock; include uwsgi_params; } }`

Chapter 11. Conclusión

11.1. Conclusión

A lo largo de la presente investigación se evidencia la importancia de que las instituciones educativas publicas cuenten con un software que les permita llevar un control eficiente del departamento de control de estudios y evaluación, así mismo ese software por razones éticas, de ahorro económico y morales debe estar basado en software libre, por ende, es vital contar con todas las herramientas disponibles para su ejecución y puesta en producción.

En esta investigación, se propuso la creación y puesta en producción del software que no solo le permite el control, sino que les permite a todos los integrantes del centro educativo el aprendizaje continuo de los procesos y el de la misma aplicación, así como les da a las instituciones la oportunidad de cumplir con decretos y leyes de la rep^ublica bolivariana de Venezuela, así su carácter de formalidad ante las instituciones del estado es de legal.

Toda la información del software "Morichal" así como de su código fuente se encuentra alojada en Github 1 y puede ser accedida por cualquier persona en cualquier lugar del planeta.