

Aplicación Web

por Estudiante Estudiante

Fecha de entrega: 27-nov-2019 09:42a.m. (UTC-0500)

Identificador de la entrega: 1222821363

Nombre del archivo: ANTIPLAGIO_VRS_19_NOV2019_REVISADO_1.docx (45.9K)

Total de palabras: 4769

Total de caracteres: 25539

CAPÍTULO 1

1. INTRODUCCIÓN

Las ventajas de Free Libre Open Source Software (FLOSS) han sido a menudo discutidas en la literatura reciente de sistemas de información; personalización, acceso libre al código fuente y los ahorros de costos son los principales aspectos positivos reportados por varias fuentes de investigación. Los beneficios que FLOSS puede traer a las organizaciones ya sean estas del sector público o privado son: reducir el fenómeno de la brecha digital, promover las economías locales, evitar dependencia del vendedor y, en una visión más amplia, atacar el dominio monopolístico en el mercado. Estos son los aspectos que en efecto aumentan la importancia de FLOSS, sin embargo, no podemos descartar el hecho de que las organizaciones en el sector público no se centran en adoptar por completo el software libre, ya que los factores para la adopción en el sector público son diferentes a los privados.

En algunos casos, las migraciones a FLOSS han logrado ser exitosas, pero esto se basa netamente en que las necesidades y requerimientos de la organización estén bien definidos previo a su evaluación.

En este contexto, se genera la pregunta, ¿Por qué el sector público no está completamente de acuerdo en adoptar una nueva solución FLOSS?

La primera razón se refiere a un aspecto endógeno de la innovación, ya que incluso las tecnologías que son beneficiosas para una comunidad pueden tardar varios años en ser ampliamente adoptadas, desplegadas y utilizadas en otra.

Pese a dicho aspecto existe otra razón. FLOSS ha sido descrito como un trastorno en la tecnología. Mientras que la innovación sostenida mejora los productos ya existentes

agregándole valor, las tecnologías disruptivas revolucionan la manera de pensar acerca de un producto.

En este caso, tres características comunes de FLOSS son:

- Software como una mercancía;
- Colaboración habilitada en red; y
- Personalización del software (software como servicio).

En primer lugar, la introducción de FLOSS plantea a la organización un radical cambio potencial, que considera a FLOSS una mercancía como un bien público impulsado por estándares abiertos y plataformas abiertas. Entonces, una clase dada de licencias FLOSS facilita la colaboración entre organizaciones que les dan la oportunidad de redistribuir software personalizado a través de una red. Finalmente, FLOSS permite un alto grado de personalización y abre la puerta a nuevas fronteras empresariales. Como tal, la adopción de FLOSS puede determinar un cambio significativo en los procesos de negocio de la organización, para adaptarlos al nuevo software.

Esto hace que la idea de adoptar FLOSS sea convincente, pero también problemática, ya que una migración a FLOSS puede implicar una gran desviación de la infraestructura organizativa tecnología de la información preexistente. Este cambio radical y la desconfianza intrínseca hacia la innovación típicamente retrasan cualquier adopción de FLOSS.

Definimos un marco teórico para la adopción de FLOSS dentro de las organizaciones, refinamos el marco teórico con los comentarios de las partes interesadas y luego construimos la herramienta para la evaluación de adopción de FLOSS. Contribuimos a la investigación en la adopción de FLOSS con:

- La definición de un marco compuesto por dimensiones, factores y subfactores para la caracterización de la adopción de FLOSS;

- La evaluación en los estudios de caso de los factores de adopción del FLOSS mediante un conjunto de métricas identificadas, y la aplicación de un protocolo de estudio empírico para evaluar los resultados del estudio intra-caso y los estudios inter-caso.

1.1. Planteamiento del problema

Las instituciones públicas y privadas del cantón Milagro tienden a adquirir software sin una previa evaluación o consultoría con respecto a las necesidades que se presentan a nivel organizacional, estas decisiones tienen como consecuencia la adopción del software incorrecto, el cual puede presentar incompatibilidad con el sistema de infraestructura tecnológica de la organización, poca flexibilidad al momento de implementarlo e interfaz poco amigable con el usuario. En este contexto, se podría evidenciar de que el software seleccionado no ha sido la mejor decisión, ya que no cumple con ciertos aspectos técnicos, organizacionales y económicos.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar una aplicación web que permita evaluar la adopción de FLOSS en instituciones públicas y privadas.

1.2.2. Objetivos Específicos

Para alcanzar el objetivo propuesto es necesario cumplir con los siguientes puntos:

- Identificar dimensiones, factores y subfactores relevantes para el proceso de evaluación de adopción de FLOSS.
- Diseñar la estructura junto al modelo conceptual de la base de datos y la aplicación.
- Desarrollo de la estructura de la aplicación mediante el lenguaje de programación Python, usando como motor de base de datos a PostgreSQL e integrando el sistema con Django.

1.3. Justificación

En la actualidad, las instituciones han empezado a apostar por nuevas tecnologías las cuales se adaptan a las necesidades de cada sector; dichas tecnologías abarcan una línea de elementos definidos como hardware y software, aquí es donde interviene FLOSS, ya que basándose en varios estudios se ha llegado a la conclusión de que posee más ventajas ante los sistemas adquiridos bajo licencias. FLOSS resulta ser de mucho provecho para aquellas instituciones que buscan expandirse y posicionarse en el mercado, ya que al ser de uso libre se produce un ahorro de inversión en comparación a los sistemas con licencias, generando así una mayor productividad dentro de la institución, junto con la oportunidad de expansión a mejoras del software sin restricciones y a su vez aprovechando los recursos de su infraestructura TI al máximo.

1.4. Marco Teórico

1.4.1. Software Libre

² Hablamos de software libre cuando queremos referirnos a la libertad que tiene un usuario para modificar, copiar, distribuir y modificar un software sin que ninguna compañía o individual pueda emprender acciones legales contra él (González, 2004, p. 9).

El error más común de concepción sobre el software libre proviene de la ambigüedad del vocablo anglosajón free, que significa tanto “libre” como “gratis”. Así pues, es habitual (y comprensiva) la confusión que la expresión free software puede generar. Aclarémosla; el software libre es un asunto de libertad y no de precio (González, 2004, p. 9).

“Es cierto que es muy habitual que el software libre sea también gratuito, pero no necesariamente tiene que ser así” (González, 2004, p. 10).

Los sistemas Open Source se definen como software libre y de código abierto que se han establecido con gran fuerza en el mercado y en el sector tecnológico, desde la fundación de Free Software Foundation (FSF) en 1984 queriendo hacerse un pequeño espacio en el ámbito de software para ordenadores; surgieron debido a que la principal función de los gobiernos era encontrar soluciones que sean eficientes y sostenibles para tecnologías de la información en el ámbito del sector público para incrementar el crecimiento económico y apoyar el bienestar social (Rossi et al., 2012, p. 135).

A pesar de ser más utilizado, FLOSS en los últimos años aún no logra posicionarse como el imperio de Microsoft, las grandes empresas no confían por completo en los sistemas abiertos, pero a pesar de la poca confianza que se le otorgue, la comunidad de software libre cree firmemente en que llegara un punto en que las empresas optaran por recursos de licencia libre (Rossi et al., 2012).

La selección del software en el sector público tiene políticas que involucran el ámbito de seguridad como también el ámbito económico. Se deben considerar aspectos de política que garanticen la seguridad, para que así la entidad pueda abrirse paso a usarla con fines de medios comerciales y sus actividades sean consideradas adecuadas dentro de los márgenes legales.

La industria del software es una de las empresas que más dinamismo tiene a nivel internacional ya que la convergencia entre la informática y las comunicaciones se ha expandido a la aplicación de software, logrando abarcar todas las actividades ya sea sociales o productivas, haciendo que las organizaciones logren tener más ventajas para poder promocionar sus productos, llegando de manera eficiente a los usuarios (Macho et al., 2017). El auge de automatización y desarrollo han contribuido a que se modifiquen las formas de trabajar de cada entidad y sobre todo la forma en la que producen sus bienes o servicios, además las organizaciones hay adoptado nuevos métodos de innovación que han mejorado sus utilidades, ya que les ha permitido llegar a más usuarios que a la final optan por adquirir sus productos o servicios a través de los medios de comunicación (Pessagno et al., 2008, p. 148).

Los usuarios de software libre van aumentando día a día, claro que no como se lo espera, ya que el incremento de estos usuarios es menor a los que usan software de licencias pagadas, las pequeñas empresas se han visto desafiadas y han tenido que idear estrategias de negocios para poder expandir sus actividades FLOSS, haciendo que se idealicen estrategias creativas mejorando el plan de negocio para poder realizar las actividades diarias de la misma manera, pero con recursos que sean más eficientes y que permitan obtener mejores utilidades, dando paso a sistemas automatizados con códigos libres. Las estadísticas de FLOSS no permiten que se distinga el ámbito económico en las diversas empresas de software libre (Macho et al., 2017).

1.4.2. Diferencia entre FLOSS y código abierto

Los dos campos políticos en la comunidad de “Free Software” son el movimiento de Software libre y el Código abierto. El movimiento del software libre es una campaña para la libertad de los usuarios de computadoras; Decimos que un programa no libre es una injusticia para sus usuarios. El campo de código abierto se niega a ver el tema como una cuestión de justicia para los usuarios, y basa sus argumentos solo en beneficios prácticos (Stallman, 2016, párr.1).

Un investigador que estudia las prácticas y los métodos utilizados por los desarrolladores de la comunidad de software libre decidió que estas preguntas eran independientes de las opiniones políticas de los desarrolladores, por lo que utilizó el término "FLOSS", que significa "Free / libre Open Source Software", para evitar explícitamente una preferencia entre los dos campos políticos. Si desea ser neutral, esta es una buena manera de hacerlo, ya que esto hace que los nombres de los dos campos sean igualmente prominentes (Stallman, 2016, párr.3).

Otros usan el término "FOSS", que significa " Free Open Source Software”, muy parecido a "FLOSS", pero en comparación es menos claro, ya que no explica que "Free" se refiere a la libertad y no a la gratuidad (Stallman, 2016, párr.4).

1.4.3. FLOSS y los modos de clasificar las empresas de software libre

La producción de software se encuentra vinculada entre los científicos y los ingenieros que laboran en centros académicos o gubernamentales para que se desarrolle como práctica de la cultura de investigación, el intercambio libre de software, la producción de estos se privatizó en la década de 1970 cuando se empezó a comercializar el pago de estos sistemas.

En gran medida, la diferencia que existe entre el Open Source y versiones de pago se establecieron por medio de organizaciones que buscaron que se privatice ciertas características de los sistemas para entidades privadas, el software en si se utiliza para corporaciones en diversos ámbitos y finalidades, desde los sistemas operativos más básicos hasta los sistemas de gestión ERP; para todos los usos empresariales existen soluciones FLOSS con funcionalidades parecidas a las soluciones propietarias (Marsan & Paré, 2013, p. 735).

Con el software propietario se crean vínculos de dependencia con proveedores de software, a partir de esto se quieren cambios y modificaciones de código, pero con FLOSS no ocurre esta dependencia ya que el código es libre y se puede contratar a cualquier persona para que se pueda desarrollar nuevas modificaciones o funcionalidades (Macho et al., 2017).

El software propietario no dispone de código fuente en las aplicaciones, por lo que se tendrá que confiar que sus desarrolladores hicieron un trabajo confiable y seguro, incluso las brechas de seguridad no se las puede conocer si el proveedor no se las especifica, esto no sucede con FLOSS ya que como se mencionó la característica de código libre hace que sea más probable detectar las vulnerabilidades y corregirlas haciendo que los sistemas sean más seguros ya que son verificados por las mismas entidades (Pessagno et al., 2008, p. 150).

A nivel de producción en las empresas y a nivel económico los términos se pueden usar indistintamente o bien como FLOSS, las ventajas de adopción pueden destacar su potencial ya que permite la difusión de software en la economía, las posibilidades que FLOSS brinda para acceder a los programas gratuitos a un costo menor que los productos que son propietarios y esto hace que se logre reducir la estructura de costos y también la dependencia tecnológica de un proveedor único (Marsan & Paré, 2013, p. 736).

FLOSS puede ser usado en varios sectores sea público o privado, aunque quienes por el momento más interactúan con él son emprendimientos pequeños y medianos con equipos de trabajo que son reducidos pero con capacidades altas, el nivel de trabajo de estas empresas es local, y su objetivo es fortalecer las redes de mercadeo que comparten con otras grandes empresas (Marsan & Paré, 2013, p. 732).

FLOSS ha ganado buena aceptación entre el área de la tecnología, y poco a poco más organizaciones lo están acogiendo como modelo de negocios. La mayoría de los proyectos que involucran a FLOSS se enmarcan en un modelo de desarrollo abierto, donde se toman datos públicos de proyectos que ofrecen la posibilidad de realizar trabajos estructurados con consideraciones de los requerimientos propios de la organización (Pessagno et al., 2008, p. 140).

Para crear una referencia hacia el avance que ha tenido el software libre en la actualidad tenemos a Moodle¹, esta es una herramienta que se basa en la web, desarrollada sobre el lenguaje PHP², con conexión a MySQL³ como base de datos. Posee un diseño sencillo de interpretar, que permite desarrollar modelos para agregar a los proyectos. Mayormente es usado en el ámbito educativo para lograr interacción con los estudiantes y hoy en día es una de las plataformas más usadas.

En diversos países del mundo las empresas públicas iniciaron la migración de su información del software propietario al software libre FLOSS, en Europa se promovió este incentivo bajo una ley establecida por la comisión europea y que a medida que pasan los años se va estableciendo con mayor fuerza (Marsan & Paré, 2013).

Debido al éxito de Apache y luego la expansión de Linux generaron más interés y confiabilidad en estos tipos de software llegando ser considerados alternativas para cambio empresarial, y ahora un modelo de negocios rentable que permitirá obtener mayor utilidades y control propio de la administración organizacional (Bouras et al., 2014, p. 238).

1.4.4. Ventajas del software libre en las empresas

La versatilidad de las herramientas de software que sirven como apoyo para los modelos de negocios que se basan en software libre poseen varias funcionalidades, por lo que existen varios softwares que dan apoyo a lenguajes de programación que intervienen en los modelos de negocios de muchas empresas, pero sobre todo lo que FLOSS ofrecerá a las organizaciones que decidan adoptarlos como suyos será:

- **Reducción de costos:** por ser la mayoría de los programas gratuitos, se convierte en gran aliado de todas aquellas organizaciones que deseen reducir costos en el ámbito de administración e informática, aplicando sistemas que le permitan hacer todas sus actividades, pero con costos menores que los que se obtendrían con Software propietario.
- **Compatibilidad con otros programas:** por lo general esta es una de las inquietudes que más se producen entre las organizaciones, saber si el software libre tendrá la capacidad de poder acoplarse con diferentes programas, por lo general se los diseña para que funcionen con las últimas mejoras de los programas que estén al despunte en la informática.
- **Escalabilidad:** permite realizar modificaciones y además obtener actualizaciones del software, esto permite que se desarrollen nuevos productos con base en el mismo y adaptarlo a las necesidades de cada organización.
- **El código es abierto, no existen secretos:** por lo cual se podrá hacer lo que sea para personalizarlo y una parte importante es que se conocerá exactamente cuáles son las vulnerabilidades que posee el sistema al momento de la inspección.

- **Se pueden hacer cambios con rapidez:** para un conocer del lenguaje en el que funciona el sistema no habrá inconvenientes para poder hacer cambios o mejoras en el sistema ya que posee un lenguaje estandarizado que todo programador podrá entender.
- **No se necesita de grandes requerimientos de hardware para funcionar:** por lo cual puede acoplarse en cualquier organización sin inconvenientes, haciendo que sea fácil y atractivo de implementar en cada empresa.

[1]: <https://moodle.org/?lang=es> [2]: <https://www.php.net> [3]: <https://www.mysql.com>

CAPÍTULO 2

2. METODOLOGÍA

La metodología forma parte del proceso de desarrollo de todo sistema o proyecto que se pretende realizar, ya que nos describe la serie de métodos y técnicas que seguiremos para alcanzar los objetivos propuestos, logrando así aumentar la eficacia al ejecutarlos y consigo, brindándonos un mejor resultado al momento de su entrega final.

2.1. Proceso de desarrollo

Según afirma (Jacobson, 2000), el modelo de proceso más adecuado para el desarrollo de software es un proceso iterativo e incremental, puesto que, a diferencia de otros modelos de proceso, como por ejemplo el modelo de cascada, permite la obtención de diversas versiones del producto software antes de la entrega final del mismo y la depuración y validación progresiva del mismo, lo que sin duda redundará en un software más satisfactorio para usuarios y cliente.

Dentro del proceso de desarrollo de sistemas se encuentran fases, las cuales detallan los pasos a seguir, priorizando sus actividades en función a los objetivos.

2.1.1. Fases de desarrollo

- **Fase de inicio**

El principal objetivo de esta fase es dar una visión aproximada del proyecto esclareciendo así los verdaderos objetivos del proyecto y el alcance que tendrá. “En esta fase se identifican y priorizan los riesgos más importantes” (Torossi, 2018).

En la fase de “Inicio” encontramos una variable esencial, la cual es el equipo con el que trabajaremos a lo largo del desarrollo, aquí es donde se detallan las especificaciones del sistema.

Personalmente nosotros hemos utilizado un equipo HP con las siguientes características:

- Sistema operativo: Windows 10
- ⁵ Procesador: Intel(R) Celeron(R) CPU N3060 @ 1.60GHz
- Memoria RAM: 4,00 GB
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64

La fase de inicio involucra una planeación que representa una idea general de las actividades y tiempos estimados para llevar a cabo el desarrollo del proyecto, por tal motivo, hemos creado un **Diagrama de Gantt** con la ayuda de Project para detallar tareas, tiempos y recursos que se ven involucrados en este proceso. Ver **Error! Reference source not found.**

Nota: Este archivo cuenta con fechas reales de los avances presentados dentro del acompañamiento de tutorías y se apega a la realidad con respecto a la fecha de entrega final del proyecto.

- **Fase de elaboración**

¹ En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los casos de uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves, bien con este prototipo, bien con otros de usar y tirar. (Martinez, 2017).

Definimos a la fase de elaboración mediante un **Caso de Uso**, el cual hace mención a los actores que intervienen o participan en el sistema, siendo estos el administrador y el usuario encuestado, donde cada uno es capaz de ejercer funciones distintas dependiendo su rol. Ver **Error! Reference source not found.**

La creación de ³ la base de datos es uno de los principales pasos dentro del desarrollo de la aplicación. Su modelo conceptual lo obtenemos con la ayuda del software DBWrench (Database Design & Synchronization Software)¹, ya que al trabajar con PostgreSQL no es posible visualizar ni manipular las tablas de forma gráfica. Ver **Error! Reference source not found.**

- **Fase de construcción**

Esta fase se centra en la construcción del producto, el desarrollo físico del software que se inicia cuando todos los casos de uso han sido implementados, los cuales aún pueden sufrir modificaciones de manera que el sistema está en pleno crecimiento y refinamiento, terminado ello se procede a realizar los códigos de programación, pruebas alfa que dan una visión del producto final. (Torossi, 2018).

Para el desarrollo del sistema utilizamos PyCharm¹ en su versión 5.0 como IDE (Entorno de Desarrollo Integrado), para posteriormente hacer uso de Python² en su versión 3.2.5 como lenguaje de programación, ya que permite la programación orientada a objetos, lo que nos ayudara a crear una conexión con Django³ de manera fácil y eficiente. Hacemos uso de Django en su versión 2.1 como base para el desarrollo de nuestra aplicación web, ya que siendo este un sistema de código abierto, resulta ser uno de los mejores frameworks si de conectividad y extensibilidad se trata.

Como sistema gestor de base de datos empleamos PostgreSQL⁴ en su versión 9.1, el cual nos ayudará a crear una conexión con PyCharm/Python para la creación de modelos y tablas que formarán parte de nuestro sistema.

Al momento de almacenar y manipular los mantenimientos hacemos uso de consultas personalizadas de Django, los cuales nos permitirán ejecutar esta tarea de la manera más óptima y eficiente.

Si hablamos de estética dentro de nuestro proyecto, debemos mencionar el uso de Bootstrap⁵ como biblioteca para los diseños de pantalla de nuestra aplicación web.

Cabe recalcar que optamos por seleccionar estas herramientas de desarrollo, por el hecho de sentirnos mucho más familiarizados al momento de trabajar con ellas.

El desarrollo del sistema se considera una de las fases más complejas y exigentes dentro de todo proyecto, ya que conlleva una larga y ardua tarea, la cual es condensar todos los conocimientos adquiridos, y ponerlos a prueba para conseguir el producto final.

Al ser un sistema de encuestas e interpretación de resultados hemos tomado como referencia el artículo científico; “A framework for evaluating organization's motivation and preparation for adopting product lines” (Rincón et al., 2018), el cual contiene información relevante para nuestro trabajo, ya que mediante métricas estadísticas analiza los factores más importantes que influyen dentro de la evaluación para la adopción de líneas de productos.

El artículo nos brinda un modelo de tablas con una matriz que nos permitirá calcular el valor de importancia de un factor y otros valores decisivos para el proceso de evaluación. Para apreciar la tabla propuesta por los autores. Ver **Error! Reference source not found.**

Según afirman (Rincón et al., 2018), la escala de respuesta para la valoración de cada factor se basa en una escala tipo Likert con un formato de cinco puntos, en donde: (5) es "completamente de acuerdo", (4) "de acuerdo", (3) "no saber", (2) "estar en desacuerdo" y (-1) "completamente en desacuerdo".

- **Fase de transición**

En esta fase es la entrega (“despliegue”) de software, donde el producto se encamina a la versión beta, las iteraciones en esta fase agregan características al software, que nacen del uso supervisado del usuario (Torossi, 2018).

El usuario puede acceder al dominio o dirección del sistema mediante el servidor de Django para iniciar sesión dependiendo su rol, ya sea este usuario o administrador.

El usuario será capaz de proceder a la evaluación y si en dado momento desea cambiar de idea en cuanto a ponderación, se le brindará la opción de re-evaluación.

El administrador posee todos los derechos y permisos del sistema, es el único que puede crear usuarios y asignar roles, como también visualizar datos estadísticos sobre las encuestas mediante reportes, analizar los movimientos que realizan los usuarios desde el módulo de auditoria, manteniendo así un mejor control; será quien tenga acceso al módulo de mantenimiento, en donde se encuentra la creación y modificación de posteriores evaluaciones, el administrador también puede formar parte de la evaluación.

- **Integración y validación del sistema**

Una vez finalizada la fase de transición se verificará que todos los componentes del sistema funcionen correctamente y que cumplan con los requerimientos establecidos, como también el correcto diseño de la base de datos y el sistema como tal, se harán las respectivas validaciones de los modelos y se comprobará el correcto funcionamiento del sistema que ayudará de manera positiva al momento de ejercer la toma de decisiones dentro de las instituciones.

[1]: <https://www.jetbrains.com/pycharm/>

[2]: <https://www.python.org>

[3]: <https://www.djangoproject.com>

[4]: <https://www.postgresql.org>

[5]: <https://getbootstrap.com>

CAPÍTULO 3

3. RESULTADOS (ANÁLISIS O PROPUESTA)

Como propuesta hacia la solución del problema, se plantea el desarrollo de una aplicación web de autoría propia cuyo nombre ha sido definido como: “Floss System Evaluator”, la cual tiene como finalidad ayudar a despejar las dudas que se presentan ante la suposición de incluir un nuevo sistema abierto dentro de una organización, esto se logrará mediante el uso de una pre-evaluación de elementos, tales como: las dimensiones, factores y subfactores que posee el sistema en cuestión. Una vez obtenida la puntuación general con respecto a los criterios de los encuestados se podrá analizar la factibilidad de aceptación y manejo del mismo, dando así lugar al proceso de adopción de FLOSS adecuado, el cual estará dirigido hacia instituciones tanto públicas como privadas.

Para determinar el grado de cumplimiento de los objetivos propuestos en este proyecto, hemos planteado un análisis de carácter descriptivo el cual nos brindará la información necesaria para conocer si dichos objetivos fueron alcanzados.

- Uno de los primeros objetivos específicos que se pretendieron alcanzar es el de Identificar dimensiones, factores y subfactores relevantes para el proceso de evaluación de adopción de FLOSS. Para alcanzar el objetivo de la identificación de dimensiones y factores se realizó la búsqueda en varias fuentes de información con temas relacionados con la adopción de FLOSS en instituciones públicas y privadas. Según la frecuencia encontrada acerca de las dimensiones y factores en artículos científicos, se realizó una matriz para codificar las mismas y construir los subfactores que son un nivel más

detallado de los factores descritos, con la finalidad de proporcionarle al usuario del sistema criterios específicos al momento de evaluar. Ver **Error! Reference source not found.**

- El segundo objetivo específico se enfoca en diseñar la estructura junto al modelo conceptual de la ⁴base de datos y la aplicación. Claramente se puede señalar como un objetivo cumplido, ya que el diseño y creación de ³la base de datos es uno de los ⁴principales pasos dentro del desarrollo de la aplicación. El modelo conceptual de la base de datos lo obtuvimos con la ayuda del software DBWrench (Database Design & Synchronization Software)¹, ya que al trabajar con PostgreSQL no es posible visualizar ni manipular las tablas de forma gráfica. Ver **Error! Reference source not found.**
 - El tercer y último objetivo específico se refiere netamente al desarrollo de la estructura de la aplicación mediante el lenguaje de programación Python, usando como motor de base de datos a PostgreSQL e integrando el sistema con Django. Esto quiere decir, la culminación y entrega final del sistema, en donde podemos afirmar que si se cumplió de la manera que se esperaba. Ver **Error! Reference source not found.**
 - Definimos como objetivo general del proyecto el “Desarrollar una aplicación web que permita evaluar la adopción de FLOSS en instituciones públicas y privadas”. Se considera como un objetivo alcanzado, ya que al cumplir con los objetivos específicos obtendremos como producto final nuestro sistema “System Floss Evaluator”, el cual nos propusimos desarrollar desde un comienzo. Ver ⁶**Error! Reference source not found.**
- Nota: En el ⁶**Error! Reference source not found.** podemos visualizar la pantalla de inicio de sesión, que a su vez se encuentra contenida en el segmento de ⁶**Error! Reference source not found.**, en donde también se exponen las demás capturas de pantalla del sistema junto a una breve descripción de su funcionalidad y uso.

- Los resultados obtenidos nos demuestran que el proceso escogido ha sido favorable para el desarrollo de destrezas y habilidades, tales como el desempeño de técnicas para el trabajo en equipo, junto con habilidades de comunicación y búsqueda de información.

[1]: <http://www.dbwrench.com/index.shtml>

CONCLUSIONES

La importancia de soluciones informáticas para evaluar distintos factores que forman parte de próximos sistemas a implementarse desempeñan un papel importante dentro de las organizaciones y crece en gran medida con el paso del tiempo, ya que resulta de mucho provecho para el área gerencial y personal responsable del departamento de tecnología de la información conocer las ventajas o desventajas que dicha solución informática pueda ofrecer, lo cual representa ser una ventaja al momento de prevenir problemas a futuro, ya que dichas áreas son las responsables de una correcta selección del software a implementarse en las instituciones..

Empleando datos reales para el análisis de evaluación y adopción de sistemas abiertos, nuestra App, “System Floss Evaluator” hace posible el ingreso de información como punto de partida (dimensiones, factores y subfactores) para generar una evaluación detallada, crear reportes y gráficos estadísticos con facilidad, contribuyendo de esta manera a una mejor toma de decisión de la solución informática a implementar.

El presente estudio puede servir para trabajos futuros y ser usado como referencia para ampliar el tema de migración y adopción de FLOSS dentro de instituciones públicas y privadas.

Aplicación Web

INFORME DE ORIGINALIDAD

4%

INDICE DE SIMILITUD

3%

FUENTES DE INTERNET

0%

PUBLICACIONES

3%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

xdocs.net

Fuente de Internet

1%

2

Submitted to BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA BIBLIOTECA

Trabajo del estudiante

1%

3

www.willydev.net

Fuente de Internet

1%

4

Submitted to Universidad Pontificia Bolivariana

Trabajo del estudiante

1%

5

yuranisdavidgalindo.blogspot.com

Fuente de Internet

<1%

6

Submitted to Universidad ESAN -- Escuela de Administración de Negocios para Graduados

Trabajo del estudiante

<1%

Excluir citas

Activo

Excluir coincidencias

< 20 words

Excluir bibliografía

Activo