BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

SCC-FCC: SISTEMA DE CONTROL DE CAMBIOS DEL SITIO WEB DE LA FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

TESIS PRESENTADA POR:

GEORGINA CERVANTES MELÉNDEZ

ASESOR:

M.C. ANA CLAUDIA ZENTENO VÁZQUEZ

Agradecimientos

Primeramente doy gracias por permitirme tener tan buena experiencia dentro de mi universidad, gracias a mi universidad por permitirme convertirme en un ser profesional, gracias a cada maestro que hizo parte de este proceso integral de formación, que como recuerdo y prueba viviente en la historia; esta tesis, que perdurara dentro de los conocimientos y desarrollo de las demás generaciones que están por llegar.

Por otra parte, también agradezco por permitirme tener y disfrutar a mi familia, gracias a mi familia por apoyarme en cada decisión y proyecto, gracias por creer en mí. No ha sido sencillo el camino hasta ahora, pero gracias a sus aportes, a su amor, a su inmensa bondad y apoyo, lo complicado de lograr esta meta se ha notado menos. Les agradezco, y hago presente mi gran afecto hacia ustedes, mi grandiosa familia.

En especial, gracias a mis padres por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí y en mis expectativas, gracias a mi madre por acompañarme, apoyarme y cuidar de mí en esta larga travesía, gracias a mi padre por siempre desear y anhelar siempre lo mejor para mi vida, gracias por cada consejo y por cada una de sus palabras que me guiaran durante mi vida.

El desarrollo de esta tesis no lo puedo catalogar como algo fácil, pero lo que sí puedo hacer, es que durante todo este tiempo pude disfrutar de cada momento, que cada investigación, proceso y proyectos que se realizaron dentro de esta, lo disfrute mucho, y no fue porque simplemente me dispuse a que así fuera, fue porque mis amigos siempre estuvieron ahí, todos aquellos que estuvieron presentes durante toda o la mayor parte de la realización y desarrollo de esta tesis, gracias a aquellos que con respeto y decencia realizaron aportes a esta, gracias a todos.

En fin, gracias a la vida por este nuevo triunfo, gracias a todas las personas que me apoyaron y creyeron en la realización de esta tesis. Gracias infinitas.

Resumen

Uno de los mayores problemas a afrontar en los proyectos es la gestión de cambios, no solo cuando éste es desarrollado por distintas personas, sino también en la organización y estructuración a nivel individual. Para solventar dicho problema, se recurre a las llamadas herramientas de control de cambios, las cuales facilitan principalmente el almacenamiento de los elementos a gestionar, la recuperación de cada uno de ellos y el registro histórico e identificación de cada una de las modificaciones realizadas en las sucesivas solicitudes de cambio ya sea en el código del proyecto o en la documentación que esta contiene.

En 2010, Tom Alby establece que un sistema de control de cambios es cualquier sistema que se haya implementado y que tenga el propósito esencial de asegurar que el proceso de realizar cambios no se realice de manera arbitraria y sin pensar, sino que es considerado cuidadosamente y finalmente aprobado por una parte responsable.

El sistema de control de cambios generalmente abarca no sólo los elementos específicos involucrados en la decisión final de aprobar, rechazar o posponer algún cambio, sino también todos los procesos que deben utilizarse. Por ejemplo, algunos de los elementos involucrados en un proceso de control de cambios que funcione correctamente pueden incluir, pero no están necesariamente limitados a, una Política de Control de Cambios previamente documentada, una Junta de Control de Cambios establecida, un núcleo establecido de Herramientas de Gestión de Cambios y, en algunos casos, un equipo de control de calidad y un equipo de gestión de activos.

En la Facultad de Ciencias de Computación existen tres sitios web que se tienen en el área de servicios de red y no existe una persona asignada solo a estas tareas, por lo que se hace necesario contar con un historial de cambios que permitan observar los cambios solicitados, generando una base de datos de conocimiento para las nuevas personas que se integran al área y realicen cambios con mayor facilidad.

De igual forma la Facultad de Ciencias de Computación también conocida por sus siglas FCC, cuenta con el Área de Servicios de Red que aloja el servidor web con el sitio principal de la facultad. Para realizar cambios y actualizaciones en el sitio web se cuenta con un grupo de alumnos de servicio social que son encargados por aproximadamente medio año. Esta situación, no permite tener una forma de trabajo ordenada y secuencial con respecto a las

solicitudes de cambios. En algunas ocasiones, y debido a causas externas, la información del sitio se pierde y es necesario revisar las solicitudes anteriores que llegan a la cuenta de correo de administradores. Lo que genera realizar búsquedas en la bandeja de entrada y revisar cada correo para validar que los cambios se mantengan. Impactando en tiempo de búsqueda y revisión del sitio.

Por lo tanto, se propuso el desarrollo de un sistema de control de cambios personalizado para la página web de la Facultad de Ciencias de Computación que permita validar y buscar por fechas los cambios, además de contar con un campo de observaciones en el que es más fácil detectar situaciones particulares de la información.

La mayor ventaja que tendrá el sistema es la base de datos de conocimientos, un campo que permitirá guardar las soluciones encontradas y aplicadas en la solución de cada solicitud requerida por parte de dirección o secretaría académica.

Índice

Lista de Tablas

Lista de Ilustraciones

Capítulo 1

Introducción

Tradicionalmente el desarrollo de un sitio web conlleva diferentes etapas desde una reunión con un cliente con el cual se debate que tipo de sitio web y que funciones necesita, el diseño y finalmente la implementación, pero ahí no se detiene el trabajo del grupo que mantiene actualizado el sitio web, también dicho sitio web debe llevar cada cierto tiempo mantenimiento, este proceso conlleva la actualización del sitio web ya sea que se actualicen o se eliminen algunos datos, dicho proceso de cambio depende del tipo de solicitud que haga el cliente.

Para actualizar el sitio web de la Facultad de Ciencias de Computación (FCC) se reciben nuevas solicitudes por correo electrónico al que todo el grupo de servicio social tiene acceso. Este proceso consiste en recepción de nuevas solicitudes de cambio, estas solicitudes son recibidas en un correo electrónico al cual todo el grupo tiene acceso. Dicho grupo cada mes decide el orden de quien atenderá cada solicitud que llegue, para así evitar que un miembro del grupo atienda más de una solicitud. Al llegar una nueva solicitud el primero en la lista atiende dicha solicitud, estas solicitudes suelen dividirse en dos tipos de cambios:

- En código: en la solicitud se especifica que parte del archivo se desea cambiar o modificar.
- En documentos: en la solicitud se especifica la actualización de documentos, imágenes o la carga nuevos documentos.

Al finalizar de atender la solicitud se pasaba el turno de atención al siguiente miembro en la lista.

Esta gestión resulta ser ineficiente, con grandes pérdidas de tiempo intentando encontrar la última modificación hecha en un archivo concreto, o en saber si un fichero ha sido modificado, cuándo y por quién. El desarrollo de software es en la actualidad algo mucho más amplio, generalmente muy dinámico y los productos generados altamente susceptibles al cambio. Ni se reduce a una labor individual, ni a un pequeño grupo de desarrolladores, sino que son muchos los desarrolladores que necesitan acceder en paralelo al mismo código fuente y realizar sus propias modificaciones. Intentar coordinar manualmente las diferentes modificaciones concurrentes sobre el mismo elemento mediante convenciones comunes es una labor tediosa e incluso puede llegar a ser imposible en cuanto el grupo de trabajo aumenta y se dispersa. Por ello se hace imprescindible el uso de herramientas que potencien la

comunicación y la coordinación entre todos ellos de manera automática y de forma más eficiente, los sistemas de control de cambios.

1.1 Conceptualización

Los sistemas de control de cambios se encargan de almacenar el historial de las modificaciones que van sufriendo los diferentes archivos de un proyecto, es decir, de gestionar los sucesivos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo manteniendo de manera estructurada los avances, retrocesos y modificaciones del software mientras está siendo desarrollado. Las ventajas de utilizar un sistema de control de cambios son innegables, no solo son eficaces en la gestión del trabajo concurrente de varios desarrolladores permitiendo manejar los diferentes cambios del proyecto, sino que permiten seguir la evolución del mismo. Cada ínfimo cambio estará registrado en el sistema con todos los datos que se puedan llegar a necesitar (qué, quién, por qué, cuando, etc.). Favorecen la colaboración entre los integrantes del equipo facilitando la integración de los cambios, encontrando así los cambios potencialmente incompatibles y ayudando a resolver tales conflictos. Sin dejar de lado la posibilidad de solucionar los errores introducidos con mayor facilidad volviendo a una versión anterior con solo elegir la deseada. No obstante lo anterior y contando con todas las ventajas que ofrece un sistema de control de cambios no se debe olvidar que por el momento ninguna herramienta es capaz de sustituir la comunicación entre los desarrolladores, y en último caso, de existir cambios incompatibles sobre la misma información, será necesaria la intervención humana para tomar la decisión definitiva.

En resumen el control o gestión de cambios en un proyecto pretende identificar, organizar y controlar las modificaciones que pueda sufrir la documentación que se almacena en el sitio web, y la estructura del mismo, por lo que se ve la necesidad de plantear un mecanismo que contribuya.

- **Administrador del control de cambios**: es una persona de interesada en el proyecto formalmente constituido que define como los entregables del proyecto y su documentación serán controlados, cambiados y aprobados. Es el encargado de atender y dar de alta las solicitudes de cambios.

- **Root**: es aquel con derechos a modificar el estatus de una solicitud y ver quien ha atendido dicha solicitud.
- **Solicitud de cambio**: es una petición que puede expandir o reducir el alcance del proyecto, modifica políticas, procedimientos, planes, modifica costos, presupuesto y la programación del proyecto. Estas solicitudes pueden ser directas o indirectas, externas o internas y legales obligatorias u opcionales. Solo las solicitudes de cambios aprobadas por el administrador a cargo serán implementadas.

1.2 Objetivos a realizar

El principal objetivo de la gestión de cambios es la evaluación y planificación del proceso de cambio para asegurar que, si este se lleva a cabo, se haga de la forma más eficiente, siguiendo los procedimientos establecidos y asegurando en todo momento la calidad y continuidad de la documentación almacenada en el sitio web.

El proceso de control de cambios se aplica de igual forma tanto en el proceso de desarrollo del sitio web y actualización de documentación almacenada en dicho sitio como en la etapa de garantía, esto con el propósito de asegurar que los cambios son controlados por método unificado y garantiza el desempeño de los procesos y correcta administración al alcance.

Los cambios pueden ocurrir durante las fases de elaboración, construcción y transición de la solución del proyecto, es decir en cualquier tiempo antes o después de su implantación y estabilización en producción.

Los posibles orígenes de los cambios puedes ser:

- Nuevas necesidades de los usuarios del sitio web.
- Reorganización, reducción o crecimiento de la solución del proyecto.
- Redefinición de los requerimientos de la solución del proyecto.
- Actualización de la documentación almacenada en el sitio web.
- Identificar la necesidad del cambio
- Realizar la solicitud de cambio.
- Evaluar solicitud.

- Si la solicitud es aceptada, se inicia el proceso de cambio solicitado.
- Se genera la orden de cambio.
- Se realiza el cambio el cual es revisado.

1.3 Facultad de Ciencias de la Computación

La Benemérita Universidad Autónoma de Puebla cuenta con una facultad destinada al estudio y desarrollo de las tecnologías, es decir, la Facultad de Ciencias de la Computación la cual cuenta con tres carreras:

1. Licenciatura en Ciencias de la Computación.
2. Ingeniería en Ciencias de la Computación.
3. Ingeniería en Tecnologías de la Información.

La facultad cuenta con cuatro edificios para la enseñanza de dichas carreras, también cuenta con los siguientes laboratorios:

- Laboratorio de Hardware.
- Laboratorio de Redes.
- Laboratorio de Base de datos.
- Laboratorio Mac.

Donde se desarrollan las clases dirigidas a dichos laboratorios. También cuenta con cuatro laboratorios con equipo de cómputo disponible, con el software que el profesor solicite para impartir y completar su clase.

En caso de que el estudiante necesite apoyo en sus estudios la facultad cuenta con una biblioteca, cursos de programación, matemáticas e incluso ingles disponibles para quien necesite mejorar su rendimiento académico.

Por último, la facultad también cuenta con sitios de esparcimiento como las palapas que se encuentran en frente del edificio C004, otras a lado del edificio C001 y otras enfrente de la cafetería, también cuenta con tres mesas de ping pong y una cancha de basquetbol.

La Facultad de Ciencias de la Computación cuenta con una página web destinada a Secretaria Académica, la cual ofrece al estudiante información sobre eventos, cursos, horarios, temarios, servicios escolares, etc. Dicha página durante ciertos periodos de tiempo necesita mantenimiento y actualización de la información que muestra, este trabajo está a cargo de

los estudiantes que aplican para el servicio social en el laboratorio Área de Servicios de Red (ASR). El manteamiento consisten en que la información que se muestra sea la correcta y que todo enlace funcione correctamente, también en caso de ser solicitado actualizar y/o eliminar la información que se muestra.

1.4 Situación actual

Actualmente en el laboratorio ASR, que se encarga del mantenimiento de la red universitaria y mantenimiento de la página web de la facultad, ha tenido dificultades para mantener un registro de los cambios realizados en dicha página, antes no se manejaba un registro donde se especificaba el cambio hecho y quien lo realizó. La única solución que se encontró en aquel momento fue realizar dicho registro en un documento de Excel en google drive, donde todos los integrantes tendrían acceso y así poder ver que se ha modificado en la página web en caso de que fuera necesario, ya sea porque el cambio realizado sea erróneo o los cambios hechos en la página ahora causen errores de visualización, etc.

1.5 Propuesta de solución

Debido al anterior punto, se realizó una reunión con el profesor encargado del laboratorio, los responsables del mantenimiento de la página web de la facultad y su servidora, donde se habló acerca de la problemática al no tener un seguimiento claro de los cambios hechos en la página web y no resguardar una copia de seguridad antes y después de realizar los cambios solicitados por parte de Secretaria Académica.
En la reunión se llegó al acuerdo de tener una página web destinada solamente al registro de las solicitudes que llegan al laboratorio, y así tener un registro de cada solicitud que ha sido atendida, y así facilitar el trabajo del profesor o los responsables a la hora de buscar errores en la página web.

Capítulo 2

Antecedentes

El sistema web es el medio más usado por todos los usuarios de Internet, constituyéndose como el sistema asociado por defecto a la idea preconcebida que se tiene sobre Internet: páginas web (Manuel Rubio, 2010). Es importante conocer el significado que conllevan para poder entender, a posteriori, cómo funciona el sistema web:

- **Web:** en inglés, telaraña, se refiere al entramado que se imagina cuando de una página se pasa a otra enlazándose entre sí todas ellas, formando una imagen como de telaraña, o web.
- **URL:** Uniform Resource Locator, o lo que quiere decir como Localizador de Recurso Uniforme. En sí es la dirección completa de un recurso que se solicita a través de Internet (y más específicamente del protocolo HTTP).
- **URI:** Uniform Resource Identifier, o Identificador de Recurso Uniforme. Lo que es la parte de la ruta dentro de la URL. La URI puede contener, a su vez, particiones, zona de consulta, y tendrá pares de clave valor separados entre sí por el signo de igualdad y entre cada par por el símbolo del ampersand.
- **Host:** del inglés máquina. Se refiere al nombre del servidor donde se alberga la URI que se solicita. El nombre debe de ser un nombre que se pueda resolver vía DNS, pudiendo ser un conjunto de letras, símbolos (válidos: guión (-) y subrayado (_)) y números: www.midominio.com
- **Port:** o puerto. Es el número en el que escucha el servidor web. No hace falta indicarlo explícitamente, a menos que sea diferente del puerto estándar, que es el 80. Se indica en formato decimal.
- **Servidor web:** es el programa que se mantiene a la escucha para atender las peticiones de los navegadores, a modo de servir las páginas web. También puede ejecutar códigos para generar páginas, imágenes y otros contenidos para cada petición. Hay servidores de pago y gratuitos, y libres, algunos ejemplos de los más usados: Apache, Cherokee y Microsoft IIS.
- **Navegador web:** es una aplicación de escritorio que permite, dando una URL, visualizar una página web, escrita en lenguaje HTML. Los navegadores web tienen la posibilidad de emplear plugins de forma que puedan: reproducir vídeos, sistemas interactivos Java, Flash, Silverlight; reproducir audio, usar hojas de estilo, o código

JavaScript…; algunos ejemplos de navegadores web más usados son: Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, Opera y Safari.

2.1 Sistemas Web en la vida cotidiana

El desarrollo de software es en la actualidad algo mucho más amplio, generalmente muy dinámico y los productos generados altamente susceptibles al cambio. Ni se reduce a una labor individual, ni a un pequeño grupo de desarrolladores, sino que son muchos los desarrolladores que necesitan acceder en paralelo al mismo código fuente y realizar sus propias modificaciones. Intentar coordinar manualmente las diferentes modificaciones concurrentes sobre el mismo elemento mediante convenciones comunes es una labor tediosa e incluso puede llegar a ser imposible en cuanto el grupo de trabajo aumenta y se dispersa. Por ello se hace imprescindible el uso de herramientas que potencien la comunicación y la coordinación entre todos ellos de manera automática y de forma más eficiente, los sistemas de control de cambios.

Los sistemas de control de cambios se encargan de almacenar el historial de las modificaciones que van sufriendo los diferentes archivos de un proyecto, es decir, de gestionar los sucesivos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo manteniendo de manera estructurada los avances, retrocesos y modificaciones del software mientras está siendo desarrollado.

Un ejemplo de una herramienta conocida que guarda un historial de los cambios realizados en un documento, es Google Drive, donde un grupo de personas pueden trabajar sobre un mismo documento en diferentes intervalos de tiempo. El documento cuenta con un historial de quien ha estado trabajando en él y que cambios ha realizado.

2.2 La importancia de los controles de cambios a los sistemas

El control de cambios es un proceso comercial que tiene como objetivo garantizar que se adopte un enfoque sistemático para realizar cambios de configuración en los sistemas y en la

documentación. Pero muchas instituciones no tienen un proceso para gestionar el cambio, lo que puede alterar los sistemas críticos de la línea de negocio. El control de cambios se facilita a través de Solicitudes de cambio (RfC), que documentan los cambios propuestos, describe por qué son necesarios, muestra que han sido probados, describe los riesgos potenciales de realizar el cambio y proporciona un plan de reversión en caso de que surjan problemas inesperados. Aquí hay tres razones por las que se debería considerar implementar el control de cambios, sin importar cuán grande o pequeño sea el proyecto.

1. **Prevenir tiempos de inactividad inesperados:** cada vez que surge un problema, ya sea con una sola PC o una aplicación distribuida basada en la nube, la primera pregunta de solución de problemas que debe hacer es "¿Qué ha cambiado?". Los sistemas de TI no solo dejan de funcionar, siempre hay una razón, y el primer lugar para comenzar a buscar es quién cambió lo que en las horas previas a la interrupción. Un cambio en un servidor podría provocar una interrupción en el servicio en otro, e incluso si solo está involucrado un solo servidor, los cambios no planificados pueden provocar interrupciones.
2. **Seguridad:** lograr la seguridad correcta es difícil, y siempre es un acto de equilibrio entre usabilidad y seguridad. Pero permitir que el personal de TI realice cambios no aprobados en los servidores puede exponer involuntariamente los sistemas y los datos confidenciales de negocios a vulnerabilidades de seguridad que de otro modo serían imposibles, o en el mejor de los casos, difíciles de explotar.
3. **Sistemas documentados:** apoyar sistemas debidamente documentados es más fácil que lidiar con una cantidad desconocida. Si sabe qué cambios se han realizado, se pueden tomar decisiones sobre cambios adicionales, y puede hacer conjeturas más informadas sobre dónde comenzar sus esfuerzos de solución de problemas en caso de que haya un problema. Incluso si sus servidores no se documentaron desde el principio, documentar cualquier cambio en el futuro lo colocará en una mejor posición para admitir esos sistemas y actualizarlos cuando sea necesario.

Es mejor prevenir que curar, la prevención de cambios no deseados es el mejor curso de acción, pero algunas cosas están fuera de nuestro control, como los cambios en el entorno del hardware o las secuencias de comandos automatizadas que aplican actualizaciones o realizan tareas de mantenimiento (Russell Smith, 2017).

2.3 Herramientas disponibles

Actualmente existen diversas herramientas para mantener actualizado un sitio web, dichas herramientas se suelen dividir en dos tipos de control, control de versiones o control de cambios que se realizan en el sitio web. En ambos tipos de control difieren en las funciones que ofrece al usuario para el proceso de administrar y actualizar ya sea proyectos o páginas web. A continuación se definirá cada tipo de control y como es que funciona.

2.3.1 Control de versiones

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

Si un desarrollador web quiere mantener cada versión diseño, se hace uso de un sistema de control de versiones (VCS por sus siglas en inglés). Dicho sistema te permite regresar a versiones anteriores de tus archivos, regresar a una versión anterior del proyecto completo, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que pueda estar causando problemas, ver quién introdujo un problema y cuándo, y mucho más. Usar un VCS también significa generalmente que si arruinas o pierdes archivos, será posible recuperarlos fácilmente. Adicionalmente, obtendrás todos estos beneficios a un costo muy bajo.

2.3.1.1 Sistemas de Control de Versiones Locales

Un método de control de versiones, usado por muchas personas, es copiar los archivos a otro directorio (quizás indicando la fecha y hora en que lo hicieron, si son ingeniosos). Este método es muy común porque es muy sencillo, pero también es tremendamente propenso a errores. Es fácil olvidar en qué directorio te encuentras y guardar accidentalmente en el archivo equivocado o sobrescribir archivos que no querías.

Para afrontar este problema los programadores desarrollaron hace tiempo VCS locales que contenían una simple base de datos, en la que se llevaba el registro de todos los cambios realizados a los archivos.

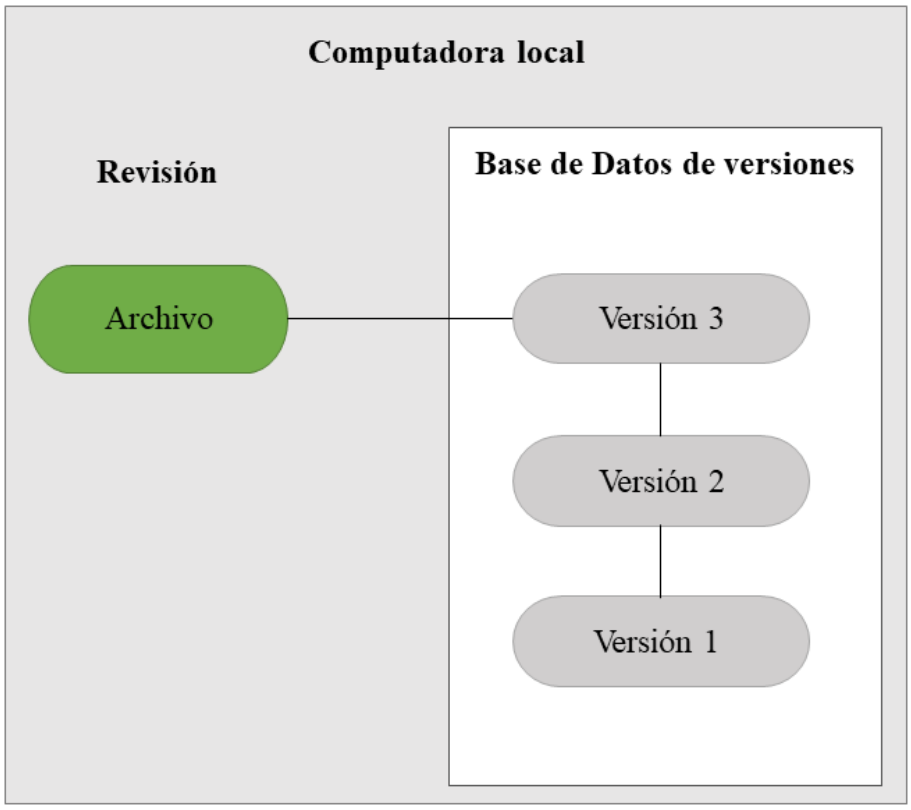


Ilustración 1 Sistema de control de versión local.

2.3.1.2 Sistemas de Control de Versiones Centralizados

El siguiente gran problema con el que se encuentran las personas es que necesitan colaborar con desarrolladores en otros sistemas. Los sistemas de Control de Versiones Centralizados (CVCS por sus siglas en inglés) fueron desarrollados para solucionar este problema. Estos sistemas, como CVS, Subversion y Perforce, tienen un único servidor que contiene todos los archivos versionados y varios clientes que descargan los archivos desde ese lugar central. Este ha sido el estándar para el control de versiones por muchos años.

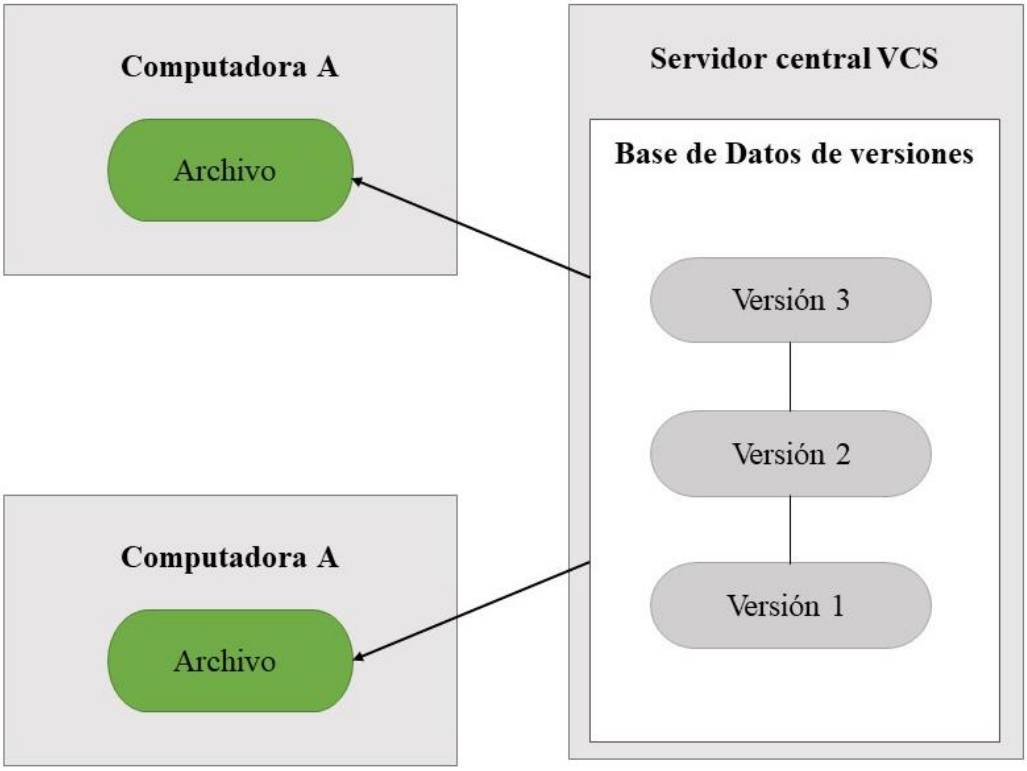


Ilustración 2. Sistema de control de versión central.

Esta configuración ofrece muchas ventajas, especialmente frente a VCS locales. Por ejemplo, todas las personas saben hasta cierto punto en qué están trabajando los otros colaboradores del proyecto. Los administradores tienen control detallado sobre qué puede hacer cada usuario, y es mucho más fácil administrar un CVCS que tener que lidiar con bases de datos locales en cada cliente.

2.3.1.3 Sistemas de Control de Versiones Distribuidos

Los sistemas de Control de Versiones Distribuidos (DVCS por sus siglas en inglés) ofrecen soluciones para los problemas que han sido mencionados. En un DVCS (como Git, Mercurial, Bazaar o Darcs), los clientes no solo descargan la última copia instantánea de los archivos, sino que se replica completamente el repositorio. De esta manera, si un servidor deja de funcionar y estos sistemas estaban colaborando a través de él, cualquiera de los repositorios disponibles en los clientes puede ser copiado al servidor con el fin de restaurarlo. Cada clon es realmente una copia completa de todos los datos.

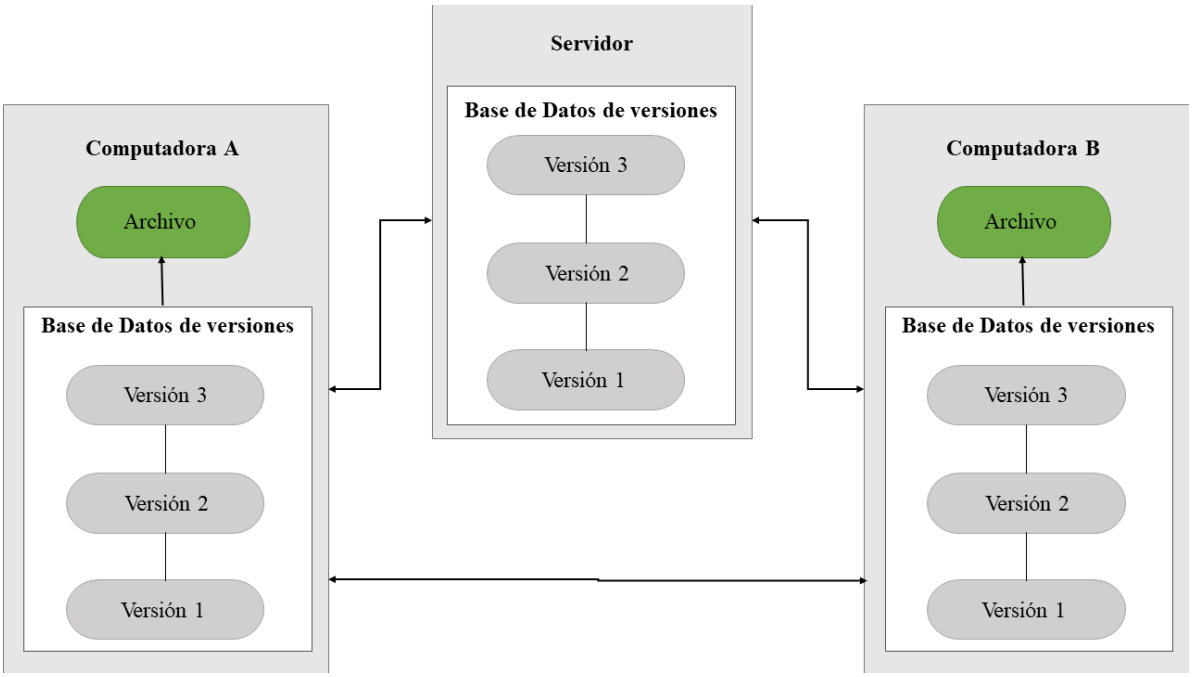


Ilustración 3. Sistema de control de versión distribuido.

Además, muchos de estos sistemas se encargan de manejar numerosos repositorios remotos con los cuales pueden trabajar, de tal forma que puedes colaborar simultáneamente con diferentes grupos de personas en distintas maneras dentro del mismo proyecto. Esto permite establecer varios flujos de trabajo que no son posibles en sistemas centralizados, como pueden ser los modelos jerárquicos.

Actualmente existen diversas herramientas que nos ayudan en el control de versiones o cambios en la documentación y código de un proyecto, a continuación se definen cinco herramientas más conocidas:

1. **Git**

   Se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo es decir a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración, es decir, control de versiones es lo que se hace al momento de estar desarrollando un software o una página web.

2. **Darcs**

Darcs es un sistema de control de la revisión distribuido creado por David Roundy; se diseñó para sustituir sistemas de control tradicionales, centralizados como CVS y Subversión. Las características claves incluyen la capacidad de elegir qué cambios aceptar de otros depósitos, interacción con otros depósitos (en el disco) locales o con depósitos remotos vía SSH, HTTP, o correo electrónico y un interfaz excepcionalmente interactivo.

3. **Microsoft Visual SourceSafe (VSS)**

   También conocido por sus siglas VSS es una herramienta de Control de versiones que forma parte de Microsoft Visual Studio aunque está siendo sustituido por Visual Studio Team Foundation Server. SourceSafe es un sistema basado en un equipo anfitrión a diferencia de la mayoría de los programas de control de versiones que son basados en Cliente-Servidor donde el repositorio de control de cambios reside en el equipo servidor y los clientes toman de allí la última versión para modificarla y posteriormente ingresarla con las modificaciones realizadas. Como en todos los programas de control de versiones, se basa en obtener una copia de trabajo ("check-out" o "desproteger"), realizar cambios sobre la copia y reingresarla al repositorio. Para lograr el acceso compartido al repositorio, VSS emplea el protocolo de archivos compartidos SMB lo cual crea algunos inconvenientes.

4. **Plastic SCM**

   Es un sistema de control de versiones distribuido propietario desarrollado por la empresa española Códice Software. Como objetivos fundamentales, Plastic trata de dar un mayor soporte al desarrollo paralelo, creación de ramas, integración (merge) de ramas, seguridad y desarrollo distribuido.

5. **IBM Rational Team Concert**

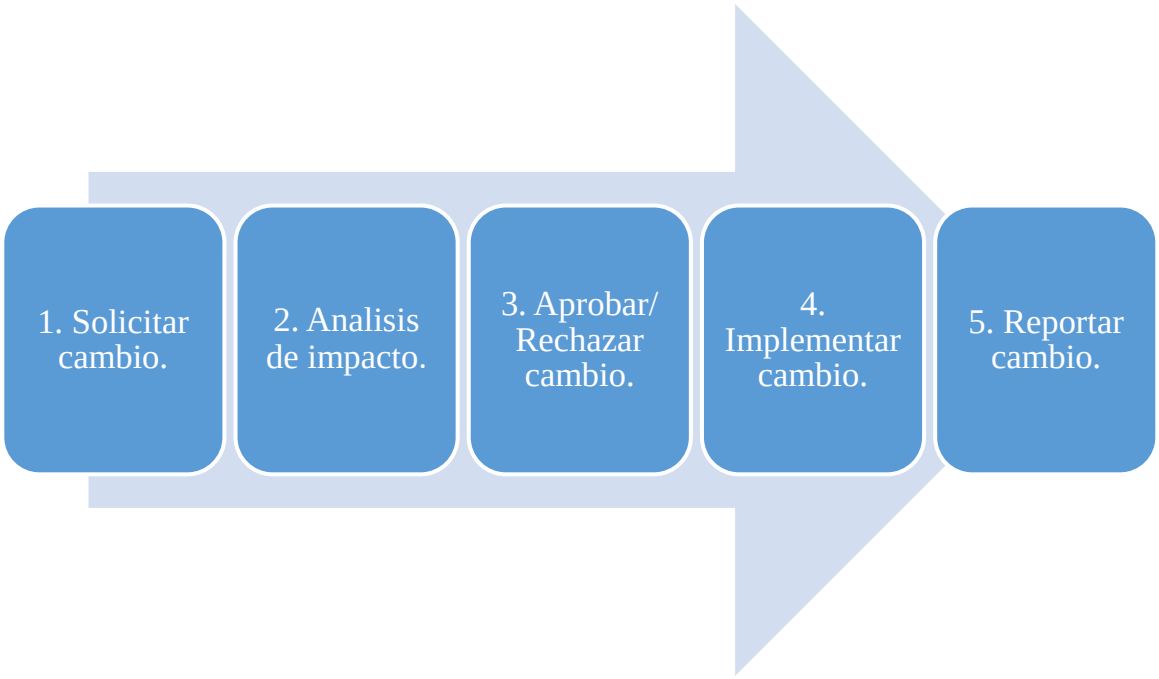
   Gestiona planes, tareas, estados de los proyectos y el vínculo fundamental entre el trabajo entregado y el flujo de trabajo. RTC tiene una flexibilidad única para adaptarse a cualquier proceso, que permite a las empresas adoptar más velozmente los ciclos de lanzamiento y administrar las dependencias entre proyectos de desarrollo, tanto pequeños como complejos. También se puede comprar como parte de la solución Collaborative Lifecycle Management, un conjunto de herramientas

integradas continuamente: IBM Rational Team Concert, IBM Rational Quality Manager e IBM Rational DOORS Next Generation.

2.3.2 Control de cambios

El control de cambios, parte de las herramientas de revisión, es una herramienta que sirve para hacer correcciones y mejorar el estilo narrativo, y además se encarga de modificar y hacer que la presentación del texto sea mejor. Esta herramienta se utiliza para que varias personas puedan revisar y/o modificar los cambios realizados en un documento hechas por otras personas. Asimismo, las herramientas de revisión sirven para hacer comentarios y/o sugerencias de los cambios realizados, e incluso se puede usar para comparar dos versiones de un mismo archivo y subrayar sus diferencias. La herramienta de control de cambios básicamente sirve para realizar cambios, y visualizar esos cambios; este también posee una herramienta que permite aceptar o rechazar un cambio realizado según convenga.
Desde principios de los años 70 se han producido un gran número de diferentes herramientas con este propósito y el objeto de este trabajo es realizar un análisis de las más significativas y una evaluación de sus características más destacables. Para ello se ha realizado una comparativa en la que se ponen en común las distintas peculiaridades de cada una de ellas, detallándola en diferentes tablas que permitan comprobar los pros y contras de cada herramienta.

Actualmente existen diversas herramientas que nos ayudan en el control de cambios o cambios en la documentación y código de un proyecto, dichas herramientas cuentan con un proceso para realizar un cambio, a continuación se muestra un diagrama de cómo funciona:



Desafíos que involucran la administración del cambio:

- Desafíos relacionados a la digitalización.
- Gestión de activos y recursos.
- Comunicación.
- Gestión de un proyecto.
- Gobierno, auditoria y análisis.
- Cambiar la forma de pensar y el enfoque hacia la estrategia básica.

Beneficios del software de gestión del cambio:

- Las herramientas de gestión del cambio ayuda a mantener un control de versión.
- Previene la modificación de un mismo objeto y por más de una persona.
- Se rastrea los cambios hechos.
- Permite deshacer los cambios.

Características del software de gestión del cambio:

- Gestión del cambio.
- Administración de incidentes.
- Administración de tareas.

- Gestión de la liberación.
- Gestión de proyecto.

El sistema de gestión de cambios sigue métodos estandarizados y procesos para incorporar los cambios al sistema. Estos sistemas realizan un análisis de extremo a extremo que a su vez es útil para tomar medidas efectivas. Todas las caracterizas mencionadas anteriormente de los sistemas de gestión ayudan a que el producto o sistema sea exitoso y también ayudan en la gestión de proyectos.

A continuación se enlistan cinco herramientas de control de cambios más populares que utilizan todas las principales organizaciones del mundo:

1. **ServiceNow.**

   Ha sido líder en el mercado ITSM durante 5 años consecutivos según Gartner's Magic Quadrant para la herramientas de gestión de servicios de TI. El producto ITSM de ServiceNow cubre la funcionalidad de gestión de cambios basada en ITIL que permite a las organizaciones recopilar solicitudes de cambio, evaluar su impacto en el sistema, planificar su implementación, preparar planes de respaldo y realizar una revisión posterior a la implementación. Ademas de eso, ServiceNow ofrece una funcionalidad que aumenta la productividad y admite iniciativas de gestión de cambios con la automatización del flujo de trabajo.

2. **FreshService.**

   Freshservice proporciona software personalizable. Es personalizable para las necesidades de TI y no TI. Este software ayuda a automatizar las tareas. Es compatible con los problemas planteados a través de un correo electrónico, un portal de autoservicio, teléfono, chat y en persona. Ayuda a gestionar los proyectos desde la planificación hasta la ejecución.

   Características:

   - La función de gestión de cambios se asegura de que el usuario esté al tanto de los cambios. Es como la comunicación entre los equipos técnicos y los usuarios.
   - Los proyectos se pueden organizar en tareas de varios niveles.
   - Puede gestionar proyectos desde la planificación hasta la ejecución.
   - Gestión de activos o inventarios.

- Administración de incidentes.
- Gestión de proyectos.
- Gestión de la liberación.
- Manejo de problemas.

3. **Change Gear.**

   ChangeGear es un repositorio basado en navegador para todos los cambios. Es una solución centralizada con una potente automatización, paneles personalizables, informes ad-hoc y flujos de trabajo multimodales. Tiene capacidades avanzadas de gestión de cambios. Tiene soluciones para DevOps, TI y negocios.

   Características:

   - Le permite registrar automáticamente los cambios realizados por el equipo de DevOps utilizando la API RESTful.
   - Esta característica le dará visibilidad completa para los cambios de TI.
   - Le permite configurar el control de cambios según sus necesidades.
   - Fácil gestión de CAB.
   - Ayuda a mejorar la comunicación entre los equipos Dev y Ops.
   - Podrá identificar los conflictos en el calendario de cambios.
   - Proporciona muchas más funciones como solución de gestión de servicios de TI.

4. **Remedy Change Management.**

   Es una plataforma de gestión de servicios de TI proporcionada por BMC Software. Facilita la facilidad de la transformación digital y ayuda a reducir los riesgos al realizar cambios organizacionales. Se puede acceder a la plataforma desde cualquier dispositivo en cualquier lugar. Se puede implementar en las instalaciones, en la nube o en un entorno híbrido.

   Características:

   - Chat en vivo.
   - Aplicación móvil.
   - Análisis de impacto.
   - Facilitar el proceso de quejas de ITIL.
   - Proporciona un tablero personalizable.

- Información basada en datos con la ayuda de informes.

El sistema proporciona las características para el administrador de la mesa de servicio y la gestión de cambios. Conexión a redes sociales para la aplicación de autoservicio. La herramienta es flexible y proporciona buenas funcionalidades como sistema de gestión de cambios.

5. **WhatFix.**

   Whatfix es una plataforma para capacitación, incorporación de empleados y soporte eficiente para el desempeño del usuario. Proporciona capacidades de capacitación como integración con LMS y servicios de asistencia compatibles con SCORM.

   Características:

   - Ayuda en el diseño de guías interactivas y no se requerirá codificación.
   - Analítica avanzada.
   - Proporciona soporte en el camino y sobre la marcha para mejorar la productividad.
   - Ofrece soporte basado en contexto.
   - Proporciona orientación para nuevas herramientas de software que resultarán en una migración de software sin esfuerzo.

2.4 Comparativa de herramientas

En esta sección se decidió analizar las herramientas orientadas al control de cambio, ya que se encontró más similitudes con las funciones que se desea desarrollar para el proyecto que se describe más adelante.

A continuación se analiza la funcionalidad específica de cinco herramientas incluyendo una exhaustiva serie de opciones y comandos, comparando cuales son los más generalizados y cuales han conseguido que algunas de las herramientas más antiguas caigan en el desuso. A la hora de elegir una herramienta de gestión de cambios es importante tener en cuenta ya no solo el precio a pagar por licencia sino también la asidua actualización de cambios llevada a cabo por el fabricante. Respecto al precio o licencia asociada, se observa que las herramientas de software libre optan generalmente por una licencia GNU General Public License que permite la libre distribución, modificación y uso del software.

Herramienta de gestión de cambio	Evaluación	Veredicto	Periodo de prueba gratis	Precio
ServiceNow	5 estrellas	Funcionalidad integral de gestión de cambios y una herramienta intuitiva de programación CAB.	No disponible	Disponible bajo demanda
FreshService	5 estrellas	Potente sistema con interfaz fácil de usar.	21 días	Blossom: $19/mes Garden: $49/mes Estate: $79/mes Forest: $99/mes
ChangeGear	5 estrellas	Un sistema personalizable que es fácil de instalar y aprender.	Demo en vivo.	Change Manager: comienza en $41 por usuario/mes Service desk: Comienza en $46 por usuario/mes Service Manager: contacto.
Remedy Change Management	5 estrellas	Una herramienta flexible con conexión a las redes sociales para la aplicación de auto servicio.	Disponible	Contactar para más detalles del precio.
WhatFix	5 estrellas	Fácil de usar y buen soporte al cliente.	No	Contactar para más detalles del precio.

Tabla 1. Antecedentes - Precio/Licencia

En la siguiente tabla contiene las características más requeridas por los desarrolladores de software en los sistemas de gestión de cambios.

	Gestión del cambio	Gestión de incidencias	Gestión de liberación	Manejo de problemas	Gestión de activos	Gestión de proyecto	Gestión de tareas

ServiceNow ITSM	Si	Si	Si	Si	Si	Si	Si
FreshService	Si	Si	Si	Si	Si	Si	Si
ChangeGear	Si	Si	Si	Si	Si	No	Si
Remedy Change Management	Si	Si	Si	No	Si	No	No
WhatFix	Si	No	No	No	No	No	Si

Tabla 2. Antecedentes - Características

Una de las grandes desventajas que presentaban muchos sistemas de control de cambios antiguos es la imposibilidad de renombrar ficheros o copiarlos a otras rutas manteniendo su historial de cambios. Como se puede comprobar en la tabla anterior, aunque en sus primeras cambios no fuera así, a fecha de hoy casi todas las herramientas han suplido ya esta carencia y actualmente sí lo permiten.

Otra característica demandada por los desarrolladores de software es la posibilidad de utilizar el estándar de codificación de caracteres UNICODE, y habiendo comprobado los últimos cambios de las herramientas sometidas a estudio se concluye que el 71% de las mismas ya cumplen con esta característica.

2.5 Conclusión

Al realizar una investigación de los diferentes tipos de herramientas que nos ayudan a administración y actualización de un proyecto o sitio web se llegó a la resolución de que las herramientas de control de cambio a pesar de cumplir con la función de la administración de los cambios realizados en un sitio web, el registro de estos cambios no es su función principal.

En referencia a la comparativa de las distintas herramientas de control de cambio, también se vio que los desarrolladores recurren a las herramientas de control de cambios que facilitan el almacenamiento de los elementos a gestionar, la recuperación de cada uno de ellos y el registro histórico e identificación de cada una de las modificaciones realizadas en los sucesivos cambios del código del proyecto.

A causa de esto, los desarrolladores de proyectos software han preferido herramientas sencillas y más fáciles de usar que los caros, completos y complejos sistemas comerciales sacados al mercado por empresas como Microsoft o IBM.

Por otra parte, la tendencia del mercado muestra un giro hacía herramientas open source con licencia GPL que ofrecen las mismas posibilidades que una herramienta comercial por la que se debe pagar un precio alto por usuario. Este giro no solo es provocado por el precio de las herramientas comerciales sino por la cercanía de los desarrolladores de sistemas de control de cambios, que ponen a disposición de los usuarios multitud de blogs y foros donde poder expresar sus disconformidades, problemas o dudas y obtener la respuesta directa de los que más saben de las herramientas concretas.

Capítulo 3

Marco teórico

Un sistema de control de cambios permite que todos los cambios documentados se coordinen adecuadamente en todos los departamentos relevantes durante todo el proceso de control de cambios. Sin un sistema de control de cambios coordinado, los usuarios podrían introducir cambios que puedan interrumpir las operaciones o deshacer los cambios enviados previamente solicitados por otra persona en un departamento diferente. Además, la implementación de un sistema de control de cambios puede mejorar la transparencia, por lo que todos los usuarios pueden ver qué cambios se han solicitado, su estado y quién es responsable de completar el flujo de trabajo.

Un sistema de control de cambios óptimo creará un sistema holístico que elimine cualquier falta de comunicación y preocupaciones interdepartamentales. Cuando se solicita un cambio, todos los usuarios pertinentes pueden ver la solicitud, determinar si existe un riesgo o no, y elaborar un plan sobre cómo se puede llevar a cabo el cambio.

Con esto en mente, se planea comparar siete herramientas estudiando diferentes características consideradas de gran importancia por los desarrolladores.

A continuación se analiza la funcionalidad específica de cada herramienta incluyendo una exhaustiva serie de tipo de capacitación, categorías e integraciones comparando entre las cinco herramientas anteriormente mencionadas.

De forma preliminar se tienen las siguientes tablas que nos dan una idea de las ventajas y desventajas de cada herramienta.

3.1 Información general

La comparativa realizada sobre las características generales de las herramientas analizadas se muestra en la Tabla 3.

Herramienta	Fundadores	Año	Precio/Licencia	Plataformas	Estado
ServiceNow	Fred Luddy David Loo Don Goodliffe Bow Ruggeri	2003	Disponible bajo demanda	SaaS iPhone iPad Android	Activo

FreshService	Girish Mathrubootham	2010	Blossom: $19/mes Garden: $49/mes Estate: $79/mes Forest: $99/mes	SaaS	Activo
ChangeGear	---	2003	Change Manager: comienza en $41 por usuario/mes Service desk: Comienza en $46 por usuario/mes Service Manager: contacto.	SaaS	Activo
Remedy Change Management	Larry Garlick	1980	Contactar para más detalles del precio.	Windows Mac SaaS iPhone iPad Android	Activo
WhatFix	Vara Kumar	2014	Contactar para más detalles del precio.	SaaS	Activo

Tabla 3. Tabla comparativa - Información general

De acuerdo con la tabla anterior, la mayoría de las herramientas suelen tener un costo que a veces depende del paquete que usuario deseé usar, y en este caso ninguna cuenta con una prueba gratis.

3.2 Tipo de Capacitación

La Tabla 4 presenta una comparativa respecto a los diferentes tipos de entrenamiento que ofrece cada herramienta, que se describen brevemente a continuación para facilitar su lectura:

- Documentación
- Webinars
- En línea
- Presencial

Tipo de capacitación	ServiceNow	FreshService	ChangeGear	Remedy Change Management	WhatFix
Documentación	Si	Si	Si	Si	Si
Webinars	Si	Si	Si	Si	Si
En línea	No	Si	Si	Si	Si
Presencial	No	Si	Si	Si	Si

Tabla 4. Tabla comparativa – Tipo de capacitación

Un punto a favor es que la mayoría de las herramientas ofrecen capacitación y ayuda, esto a través de diferentes tipos de entrenamiento.

3.3 Categorías

El análisis de las diferentes categorías que maneja cada una de las herramientas se muestra en la Tabla 5.

	ServiceNow	FreshService	ChangeGear	Remedy Change Management	WhatFix
Categorías	- Ayuda desde escritorio. - Seguimiento de problemas - Servicio de IT - ITSM	- Seguimiento de activos - Gestión del cambio - CMDB - Gestión de reclamaciones - Gestión de contratos - Ayuda desde escritorio - Seguimiento de problemas - Gestión de activos de TI - Gestión de TI - Gestión de proyectos de TI - Servicio de TI - ITSM - Conocimiento administrativo	- Gestión de activos de TI - Gestión de TI - Gestión de flujo de trabajo	- Gestión del cambio	- Plataformas de adopción digital - Micro - aprendizaje - Inducción - Formación

Tabla 5. Tabla Comparativa - Características.

3.4 Integraciones

Se muestra en este apartado una comparativa sobre las integraciones que cada herramienta soporta, a continuación se explica de forma breve cada integración y seguido se encuentra la Tabla 6 donde se comparan con cada herramienta:

- **Akkadian Provisioning Manager:** es una plataforma de automatización que lleva el aprovisionamiento, el cumplimiento, los informes SQL y la administración remota de teléfonos a otro nivel. Proporciona un enfoque táctil sin contacto o limitado para los usuarios internos y externos de forma segura, eficiente y consistente en todas las aplicaciones de Cisco Collaboration :
  - CUCM
  - CUCN
  - UCCX
  - UCCE
  - Packaged
  - CMS
  - IM&P
  - Webex Teams
- **Auvik:** es la plataforma más eficiente y rentable de administrar la infraestructura de red. Auvik es una gestión de red basada en la nube para la fuerza laboral cambiante de hoy.
  - Monitoreo automatizado.
  - Potente solución de problemas.
  - Acceso remoto seguro.
- **Device42:** es el sistema de descubrimiento sin agente más completo para IT hibrida disponible en la actualidad. Device42 puede descubrir, mapear y optimizar continuamente la infraestructura y las aplicaciones en los centros de datos y la nube, proporcionando vistas precisas de su ecosistema de IT. Device42 agrupa de manera inteligente las cargas de trabajo descubiertas por afinidades de aplicación, reduciendo drásticamente el esfuerzo requerido para crear grupos de movimiento, capturando todas las comunicaciones.

- **Dynatrace:** es una compañía de inteligencia de software que proporciona gestión de rendimiento de aplicaciones (APM), inteligencia artificial para operaciones (AIOps), monitoreo de infraestructura en la nube y gestión de experiencia digital (DEM), con productos para los departamentos de tecnología de la información y propietarios de negocios digitales de medianas y grandes empresas. Los servicios de la compañía incluyen software de gestión del rendimiento para programas que se ejecutan en las instalaciones y en la nube.
- **Grow.com:** hace que sea fácil y asequible visualizar y compartir el rendimiento de su negocio en tiempo real al unir datos de cientos de fuentes, incluidas hojas de cálculo, bases de datos y aplicaciones SaaS (como HubSpot, Quickbooks y Salesforce. Los paneles de Grow ofrecen un conjunto de herramientas y funciones potentes. Puede combinar múltiples fuentes de datos en un solo gráfico, usar las capacidades de filtro y exploración para profundizar en sus números, o emplear URLs compartidas e informes automáticos de correo electrónico (y Slack).
- **Harvest:** facilita el seguimiento del tiempo desde su escritorio, teléfono y herramientas favoritas, incluidas Asana, Trello y Basecamp. Luego, recopila estos datos de la hoja de tiempo en informes visuales intuitivos que facilitan ver en qué está trabajando su equipo, detectar proyectos antes de que excedan el presupuesto y tomar decisiones más inteligentes sobre su negocio.
- **IdentityIQ:** con SailPoint IdentityIQ, puede ver quién tiene acceso, determinar quién debe tener acceso y controlar cómo se está utilizando. La plataforma de identidad en la nube le permite controlar el acceso a cada archivo y aplicación en su entorno de TI híbrido. Y a medida que su organización crece, IdentityIQ escala con ella.
- **Igloo Software:** es un proveedor líder de soluciones digitales para el lugar de trabajo, que ayuda a las empresas a ir más allá de las intranets tradicionales a destinos digitales inspiradores que mejoran la comunicación, el intercambio de conocimientos, la colaboración y la cultura. Todas las soluciones de Igloo están 100% basadas en la nube, habilitadas para dispositivos móviles y se integran con los principales sistemas empresariales y aplicaciones en la nube de las que depende su empresa. Al centralizar toda la información, Igloo proporciona una fuente única de verdad y permite una fuerza laboral más productiva y comprometida.

- **InsideBoard:** ofrece una experiencia personalizada centrada en los empleados que sea tan atractiva como la creada para los consumidores. Puede crear un viaje simplificado e intuitivo incorporando indicadores clave de compromiso para todos sus cambios planificados en una sola plataforma. Puede decidir cómo será el éxito de sus proyectos midiendo y cuantificando la adopción y el desempeño de nuevas prácticas con nuestros Indicadores clave de éxito.
- **Leena AI:** proporciona un asistente de recursos humanos impulsado por inteligencia artificial que proporciona respuestas instantáneas a las consultas de los empleados para mejorar la experiencia de los empleados.
    - Está disponible en múltiples plataformas y canales de comunicación
    - Funciona utilizando la interfaz de usuario de plataforma estándar para una confusión mínima y la mejor experiencia de usuario.
    - Puede comunicarse con varios empleados al mismo tiempo.
    - Elimina la espera entre hacer una pregunta y recibir una respuesta.
    - El motor de inteligencia artificial de Leena permite interacciones personalizadas con cada empleado.
    - Emplea las mejores características de seguridad de su clase para garantizar que sus datos estén a salvo de miradas indiscretas.
- **LiveHelpNow Suite:** ofrece todos los canales de soporte en una solución: chat en vivo, SMS para chatear, chat bots, gestión de correo electrónico, gestión de base de conocimiento, gestión de llamadas VoIP, análisis de centros de contacto.
- **ManageEngine Desktop Central:** es una solución unificada de gestión de end points, que se encarga de la gestión de la movilidad empresarial (incluidas todas las funciones de gestión de aplicaciones móviles y gestión de dispositivos móviles), así como la gestión de clientes para una gama diversificada de endpoint: dispositivos móviles, computadoras portátiles, computadoras, tabletas, servidores máquinas, etc. Con ManageEngine Desktop Central, los usuarios pueden automatizar sus rutinas regulares de administración de escritorio, como distribuir software, instalar parches, administrar activos de TI, crear imágenes y desplegar SO, y más.
- **Matillion:** al trabajar en la nube, se reduce la complejidad que implica mover grandes cantidades de datos. Permite procesar mil millones de filas de datos en quince minutos

y pasar del lanzamiento a vivir en solo cinco. Las empresas modernas que buscan una ventaja competitiva deben aprovechar sus datos para obtener mejores conocimientos empresariales. Permite que el viaje de datos al extraer, migrar y transformar sus datos en la nube, lo que le permite obtener nuevos conocimientos y tomar mejores decisiones comerciales.

- **Microsoft Exchange:** lo ayuda a colaborar en sus documentos críticos y le brinda una bandeja de entrada enfocada que prioriza mensajes importantes y se adapta a su estilo de trabajo, para que pueda hacer más cosas más rápido. Obtiene acceso a una bandeja de entrada más personalizada con funciones útiles y una forma más inteligente y organizada de ver e interactuar con el correo electrónico.
- **Microsoft SharePoint:** potencia el trabajo en equipo con sitios de equipo dinámico y productivo para cada equipo, departamento y división de proyecto. Se puede:
  - Compartir archivos, datos, noticias y recursos.
  - Personalizar su sitio para optimizar el trabajo de su equipo.
  - Colaborar sin esfuerzo y de forma segura con los miembros del equipo dentro y fuera de su organización, en PC, Mac y dispositivos móviles.
  - Desarrollar cohesión e informe a sus empleados en toda su intranet.
  - Impulsar la eficiencia organizativa al compartir recursos y aplicaciones comunes en sitios web y portales.
- **Omnium Lite:** conjunto de herramientas de automatización de gestión de entornos de prueba y DevOps que atiende a la captura, organización, determinación de patrones de programación e informes en todos los entornos de TI de prueba y no producción. Resuelve el problema de reservar, programar, solicitar y administrar entornos de prueba, entornos de desarrollo, entornos de carga / rendimiento, no producción y todo tipo de entornos de TI de manera eficiente, negando la necesidad de usar hojas de cálculo engorrosas u otras herramientas inadecuadas
- **SaltSlack:** es una plataforma inteligente de automatización de TI que puede administrar, proteger y optimizar cualquier infraestructura, en las instalaciones, en la nube o en el borde. Se basa en un motor de automatización único y potente basado en eventos que detecta eventos en cualquier sistema y reacciona de manera inteligente

ante ellos, lo que lo convierte en una solución extremadamente efectiva para administrar entornos grandes y complejos.

- **Sensu:** es la solución preparada para el futuro para el monitoreo de múltiples nubes a escala. Sensu permite a las empresas automatizar sus flujos de trabajo de monitoreo y obtener una visibilidad profunda de sus entornos de nubes múltiples.
- **Yammer:** es un servicio de red social empresarial freemium utilizado para la comunicación privada dentro de las organizaciones. El acceso de una red de Yammer está determinado por el dominio de Internet del usuario, de modo que sólo las personas con direcciones de correo electrónico aprobadas pueden unirse a sus respectivas redes.
- **eG Entreprise:** con eG Enterprise, puede medir la experiencia digital de sus usuarios, obtener una visibilidad profunda del rendimiento de toda la pila de entrega de aplicaciones, desde el código hasta la experiencia del usuario y desde el centro de datos hasta la nube, desde un solo panel de vidrio, correlacionar el rendimiento entre dominios e identifiqua la causa raíz de los problemas de forma proactiva.

	ServiceNow	**FreshService**	**ChangeGear**	**Remedy Change Management**	**WhatFix**
Akkadian Provisioning Manager	Si	No	No	No	No
Auvik	Si	No	No	No	No
Device42	Si	Si	No	No	No
Dynatrace	Si	No	No	No	No
Grow.com	No	Si	No	No	No
Harvest	No	Si			
IdentityIQ	Si	No	No	No	No
Igloo Software	Si	No	No	No	No
InsideBoard	Si	No	No	No	No
Leena AI	Si	No	No	No	No
LiveHelpNow Suite	Si	No	No	No	No
ManageEngine Desktop Central	Si	No	No	No	No
Matillion	Si	No	No		
Microsoft Exchange	Si	No	No	No	No

Microsoft SharePoint	No	No	Si	No	No
Omnium Lite	Si	No	No	No	No
SaltStack	Si	No	No	No	No
Sensu	Si		No		
Yammer	No	Si	No	No	No
eG Enterprise	Si	No	No	No	No

Tabla 6. Tabla comparativa - Integraciones.

En resumen, la gestión de cambios es uno de los mayores problemas a afrontar en los proyectos software. Para solventar este problema los desarrolladores recurren a las herramientas de control de cambios que facilitan el almacenamiento de los elementos a gestionar, la recuperación de cada uno de ellos y el registro histórico e identificación de cada una de las modificaciones realizadas en los sucesivos cambios del código del proyecto.

3.6 ¿Qué se seleccionó para el desarrollo del sistema?

El control de cambios es uno de los mayores problemas a afrontar en los proyectos software. Para solventar este problema los desarrolladores recurren a las herramientas de control de cambios que facilitan el almacenamiento de los elementos a gestionar, la recuperación de cada uno de ellos y el registro histórico e identificación de cada una de las modificaciones realizadas en los sucesivos cambios del código del proyecto o en los documentos que contiene.

Se ha presentado un estudio de las principales herramientas para realizar la gestión de cambios en proyectos software, analizando tanto sus características como sus funcionalidades y carencias con el fin de comprobar cuáles serían las funcionalidades esenciales para la creación de una herramienta idónea que satisfaga las necesidades reales de los equipos de proyecto.

En relación con la facilidad de uso, los desarrolladores valoran positivamente la disponibilidad de diferentes interfaces, no solo a nivel web y de escritorio, sino plugins para la integración de la misma con los entornos de desarrollo habituales. Las nuevas herramientas facilitan desde el primer instante plugins para Eclipse y/o Visual Studio que hacen su interfaz amigable y lo que es más importante, conocida para el desarrollador.

De acuerdo a las funciones anteriormente explicadas se realizó una reunión con el profesor encargado del laboratorio, los responsables del mantenimiento de la página web de la facultad y su servidora, donde como objetivo de facilitar el registro de las solicitudes que llegan al laboratorio y son atendidas, se busca un sistema control de cambios personalizado con las siguientes funciones:

- Un sitio donde los todos los responsables tengan acceso.
- Un formulario, para poder obtener una base de datos, que contendrá las solicitudes que llegan al laboratorio. Este formulario recabara la solicitud y los cambios realizados en la página web de la facultad.
- Poder visualizar las solicitudes ya realizadas y verificar información en caso de ser necesario.
- Poder visualizar quien realizo dicho cambio y cuantas solicitudes ha atendido.
- Poder saber en qué estado se encuentran las solicitudes ya registradas.
- Realizar cambios o eliminar una solicitud ya registrada en la base datos.

Con esto también se especificó las acciones que cada usuario podría realizar:

- Profesor:
    - Ver solicitudes.
    - Editar solicitudes.
    - Eliminar solicitudes.
- Estudiante:
    - Registrar nueva solicitud.

Capítulo 4

Marco metodológico

4.1 Metodología

Para el desarrollo del sistema de control de cambios se empleó la metodología Scrum, a continuación se describe en qué consiste dicha metodología. Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, y obtener el mejor resultado posible de un proyecto.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

El proceso de un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de retroalimentación del producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite. El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en Scrum son las siguientes:

1. **Planificación de la iteración**
   - **Selección de requisitos**: Se realizará entrevista con los encargados de actualizar la información en el sitio web, a fin de detectar necesidades que podrá ayudarnos al desarrollo del sistema. Sobre todo en el diseño del formulario de cambios realizados.
   - **Planificación de iteración:** Se elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados.
2. **Ejecución de la iteración**

   Cada día se realiza una reunión de sincronización, normalmente delante de un tablero físico o pizarra, el equipo inspecciona el trabajo que el resto está realizando.
3. **Inspección y adaptación**

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

- **Revisión (demostración):** Se presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo.
- **Retrospectiva:** Se analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad.

4.2 Implementación

Ahora se muestra un diagrama general de que es lo que conforma y el funcionamiento del sistema de control de cambios.

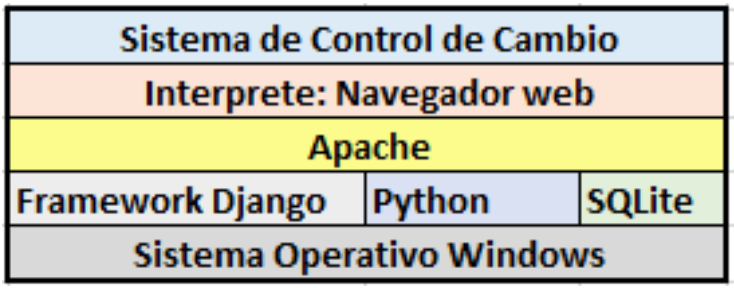


Ilustración 4. Diagrama general – Sistema de control de cambios (1)

En los siguientes puntos se describirá cada segmento y cuál es su función dentro del sistema.

4.3 Software

De acuerdo a la ilustración anterior el sistema de control de cambios se conformara de diferentes tecnologías: framework Django, lenguaje programación Python, base de datos SQLite en base al sistema operativo Windows, a continuación se detalla cada tecnología.

4.3.1 Framework Django

En la implementación del sistema, se decidió hacer uso del framework Django, el cual es un framework de desarrollo web de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Elaborado por desarrolladores experimentados, se encarga de gran parte de la

molestia del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda.

Incluye docenas de extras que se pueden usar para manejar tareas comunes de desarrollo Web. Django se encarga de la autenticación del usuario, administración de contenido, mapas de sitio, retroalimentaciones RRS, y muchas más tareas de manera inmediata.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes. Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

Características principales:

- **Python:** Para mí esto es lo mejor de todo y también es lo que se lleva "casi" todo el crédito a la definición de Django. "Django es un framework web de alto nivel" eso está muy claro en la definición, pero ahora agregaremos algo, "Django es un framework web de alto nivel escrito en Python". Gracias a esto Django hereda todas las características y facilidades que nos da Python, entre ellas escribir código bastante fácil de entender, y sobre todo te permite desarrollar aplicaciones muy rápidas y potentes.
- **Rapidez:** Django nació en un ambiente periodístico, donde se subían noticias muy rápido, y como los desarrolladores no pudieron estar a ese ritmo decidieron crear algo que sí lo haga, y así fue como nace Django, es por eso que ha sido estructurado de tal manera que tus aplicaciones web se crean muy rápidas.
- **DRY:** Django utiliza esta filosofía para no crear bloques de código iguales y fomentar la reutilización del mismo.
- **Administrador:** Django es el único framework que "por defecto" viene con un sistema de administración activo, listo para ser utilizado sin ningún tipo de configuración.
- **ORM:** Para resumir esto, tómalo como una herramienta que te permite realizar consultas SQL a la Base de Datos, sin utilizar SQL.

4.3.2 Python

Python es un lenguaje de programación interpretado, por lo que funciona en cualquier tipo de sistema que integre su interpretador. A parte de esta ventaja, Python nos ofrece dialectos como el ya conocido Jython, que se utiliza para escribir en Java. Python no sólo es multiplataforma y multiparadigma, sino que también nos servirá para desarrollar cualquier tipo de vía, como por ejemplo web o móvil. Para que esto se lleve a cabo, este lenguaje de programación cuenta con frameworks de gran calibre, los cuales auxilian desde el desarrollo web, hasta el desarrollo de juegos o algoritmos científicos de cálculos avanzados.

Python se encuentra en multitud de aplicaciones y servicios que usamos habitualmente. Ostenta una gran lista de usuarios de gran calibre como Google, YouTube o Facebook, los cuales utilizan este lenguaje de programación. Poco a poco Python va ganando territorio y, entre los entendidos, se ha convertido en uno de los lenguajes más solicitados y, sobretodo, más esenciales del momento. Esto nos demuestra que programar en Python es la opción más viable y efectiva que hay ahora mismo en el mercado.

La sintaxis que nos ofrece este lenguaje de programación es una de sus características más notorias. En Python, un bloque de código interno como puede ser un if, se crea a través de indentaciones, lo que fuerza al desarrollador a indentar su código fuente garantizando una legibilidad notoria. Otras de sus funciones son las de reducir el uso de caracteres como =, {,} entre otros, y de ser capaz de escribir un for que testee una determinada secuencia.

4.3.3 SQLite

SQLite es una biblioteca escrita en leguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado. Actualmente es utilizado en gran cantidad de aplicaciones incluyendo algunas desarrolladas como proyectos de alto nivel.
Características principales:

- SQLite es un sistema completo de bases de datos que soporta múltiples tablas, índices, triggers y vistas. No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro. El formato de la base de datos es multiplataforma e indistintamente se puede utilizar el mismo archivo en sistemas de 32 y 64 bits.
- La base de datos se almacena en un único fichero a diferencia de otros DBMS que hacen uso de varios archivos. SQLite emplea registros de tamaño variable de forma tal que se utiliza el espacio en disco que es realmente necesario en cada momento.
- El código fuente está pensado para que sea entendido y accesible por programadores promedio. Todas las funciones y estructuras están bien documentadas.
- Existe un programa independiente de nombre sqlite que puede ser utilizado para consultar y gestionar los ficheros de base de datos SQLite. También sirve como ejemplo para la escritura de aplicaciones utilizando la biblioteca SQLite.

Algunas de sus ventajas son:

- **Tamaño:** SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- **Rendimiento de base de datos:** SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- **Portabilidad:** se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- **Estabilidad:** SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- **SQL:** implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo sub-consultas, generación de usuarios, vistas y triggers.
- **Interfaces:** cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, Tcl, Groovy, Qt ofrece el plugin sqlite, etc.
- **Costo:** SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

4.3.4 Apache

Apache es un software de servidor web gratuito y de código abierto con el cual se ejecutan el 46% de los sitios web de todo el mundo. El nombre oficial es Apache HTTP Server, y es mantenido y desarrollado por la Apache Software Foundation.

Permite a los propietarios de sitios web servir contenido en la web, de ahí el nombre de "servidor web". Es uno de los servidores web más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995.

Cuando alguien quiere visitar un sitio web, ingresa un nombre de dominio en la barra de direcciones de su navegador. Luego, el servidor web envía los archivos solicitados actuando como un repartidor virtual.

¿Cómo funciona?

Su trabajo es establecer una conexión entre un servidor y los navegadores de los visitantes del sitio web (Firefox, Google Chrome, Safari, etc.) mientras envían archivos entre ellos (estructura cliente-servidor). Apache es un software multiplataforma, por lo cual funciona tanto en servidores Unix como en Windows.

Cuando un visitante quiere cargar una página de tu sitio web, por ejemplo la página de inicio o tu página "Acerca de nosotros", su navegador le envía una solicitud a tu servidor y Apache le devuelve una respuesta con todos los archivos solicitados (texto, imágenes, etc.) El servidor y el cliente se comunican a través del protocolo HTTP y Apache es responsable de garantizar una comunicación fluida y segura entre las dos máquinas.

Capítulo 5

Diseño y desarrollo

Se presenta el funcionamiento del sistema de control de cambios, donde tenemos a un usuario que ingresa al sistema a través de un navegador web, dicha interfaz está programada en el framework Django el cual conecta una base datos (SQLite) la cual recibe, procesa y envía respuesta de las solicitudes que realiza el usuario.

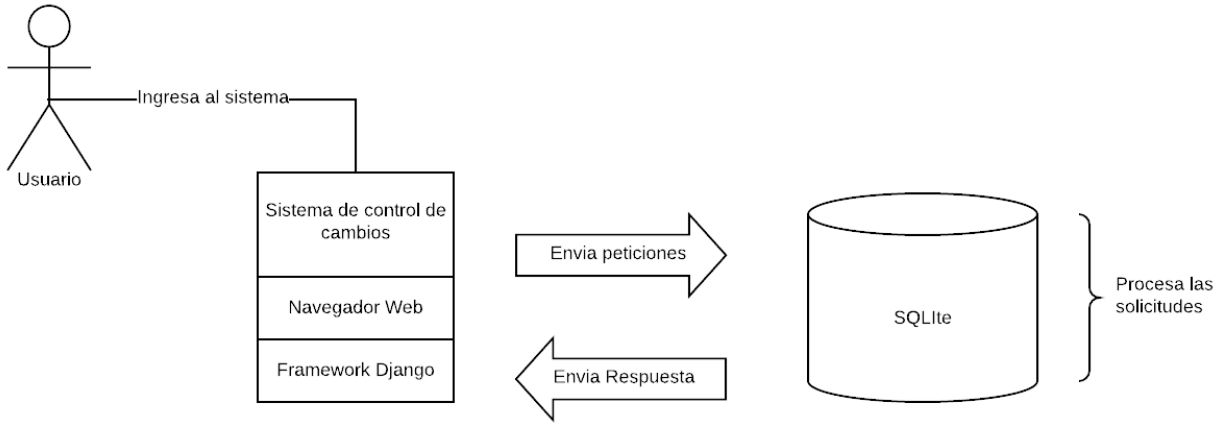


Ilustración 5. Diagrama general - Sistema de control de cambios (2)

5.1 Estructura del sistema

En el siguiente diagrama de flujo se describe la secuencia básica para la atención de una solicitud de cambio, que reciben en el laboratorio:

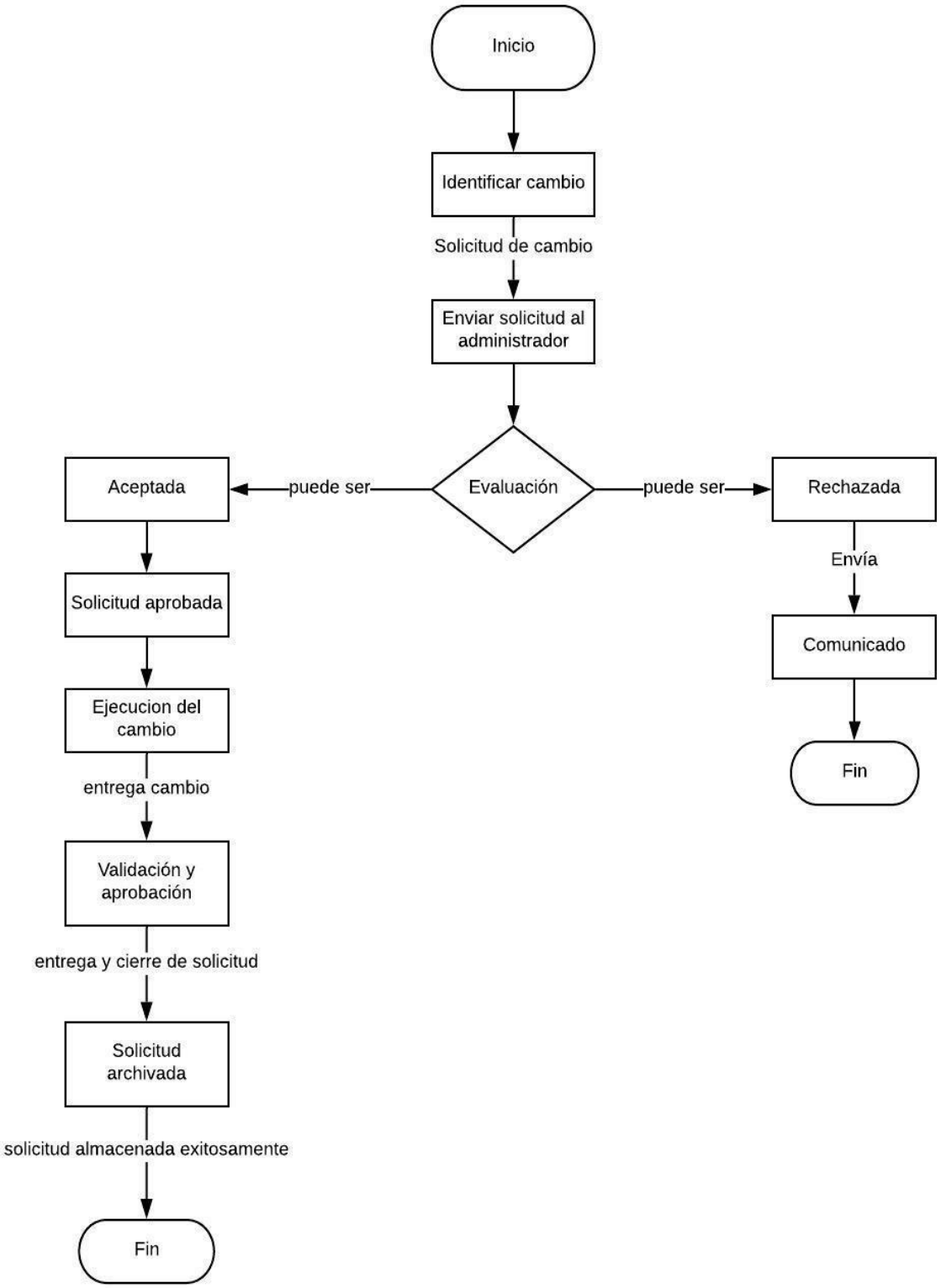


Ilustración 6. Diagrama de flujo - Proceso para la atención de una solicitud

El proceso inicia cuando se identifica un cambio necesario ya sea en la página web o en la documentación alojada en el mismo. Después se envía un correo especificando que tipo de cambio se necesita y adjuntando la documentación necesaria para realizar el cambio, el administrador al recibir la solicitud se evalúa, en caso de ser rechazada se envía un comunicada al solicitante, por el contrario si la solicitud es aceptada, se inicia la ejecución del cambio, al finalizar se entrega el cambio y se espera la validación y aprobación del solicitante. Finalmente el administrador da de alta la solicitud para llevar un registro de las solicitudes que se han atendido, se están atendiendo o están en espera.

5.2 Descripción de Usuarios

El sistema cuenta con dos tipos de usuario, a continuación describiré:

1. **Root:** este usuario es manejado por el o los profesores encargados del laboratorio ASR.
2. **Administrador:** este usuario es universal para los responsables del laboratorio de ASR, ellos atienden las solicitudes para realizar cambios en el sitio web de la facultad.

Cada usuario cuenta con un nombre de usuario y contraseña único.

5.3 Permisos

El sistema al contar con dos tipos de usuario, cada uno tienen diferentes permisos sobre la base de datos, a continuación se explicara los permisos de cada usuario:

1. **Root:**
   - Ingresar al sistema con su usuario y contraseña única.
   - Modificar estatus de las solicitudes, es decir, si en una solicitud el estatus esta errado, deberá modificar dicho estatus de la solicitud.
   - Ver la actividad que realiza cada usuario, esto consiste en:
     - Ver cuantas solicitudes ha atendido.
     - Cuáles son las solicitudes que ha atendido.

2. **Administrador:**
    - Ingresar al sistema con su usuario y contraseña única, solo existirá una cuenta única de administrador.
    - Dar de alta una solicitud.

5.4 Casos de Uso

A continuación se describirán los casos de uso del sistema:

CU-01	**Dar de alta una solicitud**	
Versión	1.0 (01/10/2019)	
Dependencias	Atender una nueva solicitud.	
Precondición	El usuario se debe identificar como administrador.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso, cuando el usuario administrador haya atendido una nueva solicitud, y esta no haya sido dada de alta en el sistema.	
Secuencia normal	**Paso**	**Acción**
	1	El administrador ingresa su usuario y contraseña, el sistema verifica que los datos sean correctos.
	2	El administrador llena el formulario con los datos de solicitud que atendió.
	3	El administrador al terminar de llenar el formulario, da clic en el botón “Guardar”.
	4	El sistema muestra el siguiente mensaje “La solicitud ha sido archivada exitosamente”.
	4.1	En caso de que el administrador quiera dar de alta otra solicitud, deberá dar clic en el botón que aparece debajo del mensaje.
	5	Finalmente si el administrador ha terminado, deberá cerrar sesión.
Post condición	El usuario administrador habrá atendido cierto número de solicitudes y el sistema tendrá actualizado la base de datos.	
Excepciones	**Paso**	**Acción**
	1	Si el usuario ingreso erróneamente sus datos, el sistema le muestra un mensaje de error ya sea en el usuario o contraseña.
Comentarios	Ninguno	

CU-02	**Modificar solicitud**	
Versión	1.0 (01/10/2019)	
Dependencias	Solicitud con el estatus erróneo.	
Precondición	El usuario se debe identificar como root.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso, cuando al usuario root le hayan avisado que cierta información de la solicitud es errónea, en este caso el estatus.	
Secuencia normal	**Paso**	**Acción**
	1	El root ingresa su usuario y contraseña, el sistema verificara que los datos sean correctos.
	2	El usuario root seleccione el botón “Modificar solicitud”.
	3	El sistema muestra una tabla con las solicitudes en la base de datos.
	4	El usuario root seleccione la solicitud que desee modificar.
	5	El sistema le mostrar el asunto, contenido y estatus de la solicitud que selecciono, solo podrá modificar el estatus en caso de que este erróneo. Al terminar deberá dar clic en el botón “Guardar cambios”.
	6	El sistema le volverá a mostrar la tabla con todas las solicitudes en la base datos.
	7	Finalmente si el usuario root termino de modificar las solicitudes, deberá cerrar sesión.
Post condición	La base de datos estará actualizada.	
Excepciones	**Paso**	**Acción**
	1	Si el usuario ingreso erróneamente sus datos, el sistema le mostrara un mensaje de error ya sea en el usuario o contraseña.
Comentarios	Ninguno	

CU-03	**Ver actividad**	
Versión	1.0 (01/10/2019)	
Dependencias	La base de datos deberá contener registros.	
Precondición	El usuario se debe identificar como root.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando al usuario root desee verificar la actividad de usuario administrador.	
Secuencia normal	**Paso**	**Acción**
	1	El root ingresa su usuario y contraseña, el sistema verificara que los datos sean correctos.
	2	El usuario root seleccione el botón “Ver actividad”.
	3	El sistema muestra una tabla con los datos de quien y cuantas y cuales solicitudes ha atendido.
	4	En caso de haber terminado la revisión, el usuario root podrá cerrar sesión.
Post condición	El usuario root podrá verificar quienes han estado atendiendo las solicitudes.	
Excepciones	**Paso**	**Acción**
	1	Si el usuario ingreso erróneamente sus datos, el sistema le mostrara un mensaje de error ya sea en el usuario o contraseña.
Comentarios	Ninguno	

5.5 Diseño de Base de datos

Para explicar la base de datos que utiliza el sistema, se realizó un diagrama entidad-relación, donde los elementos son los siguientes:

Entidad: Solicitud_Detalle y Usuario_Detalle.

Relación: la relación entre las dos entidades es que un usuario puede atender de una a muchas solicitudes, mientras que muchas solicitudes pueden ser solo atendidas por una persona.

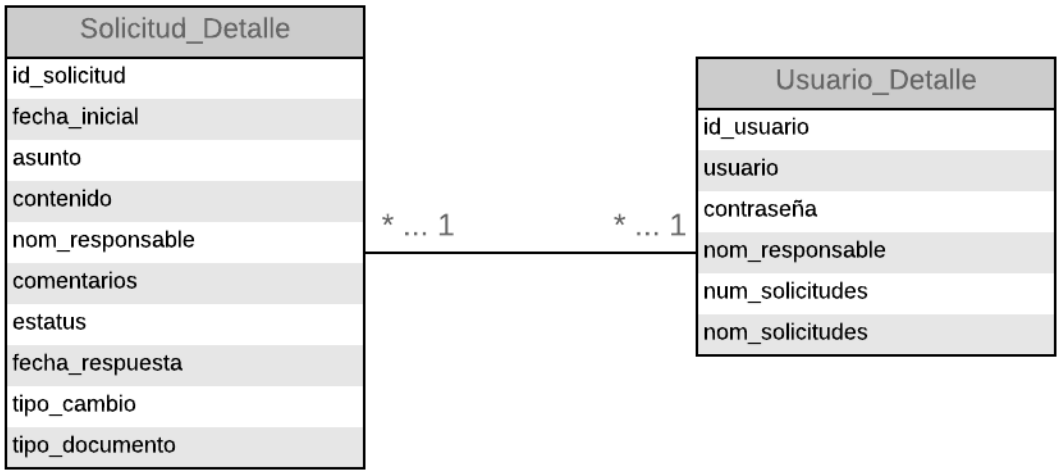


Ilustración 7. Diagrama Entidad-Relación - Diseño de la Base de Datos

En base a una reunión con los estudiantes a cargo del mantenimiento de la página web de la facultad, se llegó a un acuerdo de cuáles eran los campos más importantes a la hora del registro de una nueva solicitud, los cuales se explican más adelante. De igual forma, se propuso los campos que debería tener el usuario del estudiante los cuales son:

- Usuario
- Contraseña
- Nombre del responsable
- Número de solicitudes
- Nombre de las solicitudes que ha atendido.

Esto para poder ver su actividad mientras esta de servicio en el mantenimiento de la página web de la facultad.

5.6 Formulario para el control de cambio

De acuerdo al proceso de atención de una solicitud, se ha propuesto el siguiente formulario para llevar un registro de las solicitudes atendidas. Este formulario contendrá los siguientes enunciados:

- Fecha en que se solicitó el cambio. (con el siguiente formato dd/mm/yyyy)

- Asunto (del correo).
- Contenido (del correo).
- Quién realizó el cambio (en caso de haber más responsables de la ejecución del cambio se separan los nombres por una coma).
- Comentarios (el responsable explica como realizó el cambio).
- Estatus de la solicitud (Rojo: solicitud urgente, Naranja: solicitud pendiente, Verde: solicitud atendida).
- Fecha de respuesta (día en que se atendió la solicitud, con el siguiente formato dd/mm/yyyy).
- Tipo de cambio: código o archivo.

Si selecciona código, deberá especificar en la sección de "comentarios" que fue lo que cambio.

En caso de seleccionar archivo, deberá especificar en qué tipo de archivo realizo el cambio:

- Acta
- Horario
- Protocolo
- Titulación
- Imagen(carrusel)

Al finalizar dicho formulario y dar clic en el botón "Guardar", se muestra un mensaje confirmando el registro de la solicitud atendida, también tiene la opción de poder registrar otra solicitud que haya atendido.

5.7 Vistas

A continuación, se muestra la propuesta para el diseño del sistema de control de cambios de la Facultad de Ciencias de la Computación:

Ingreso al Sistema: tanto el usuario root como administrador deberán ingresar sus usuarios para poder acceder al sistema.

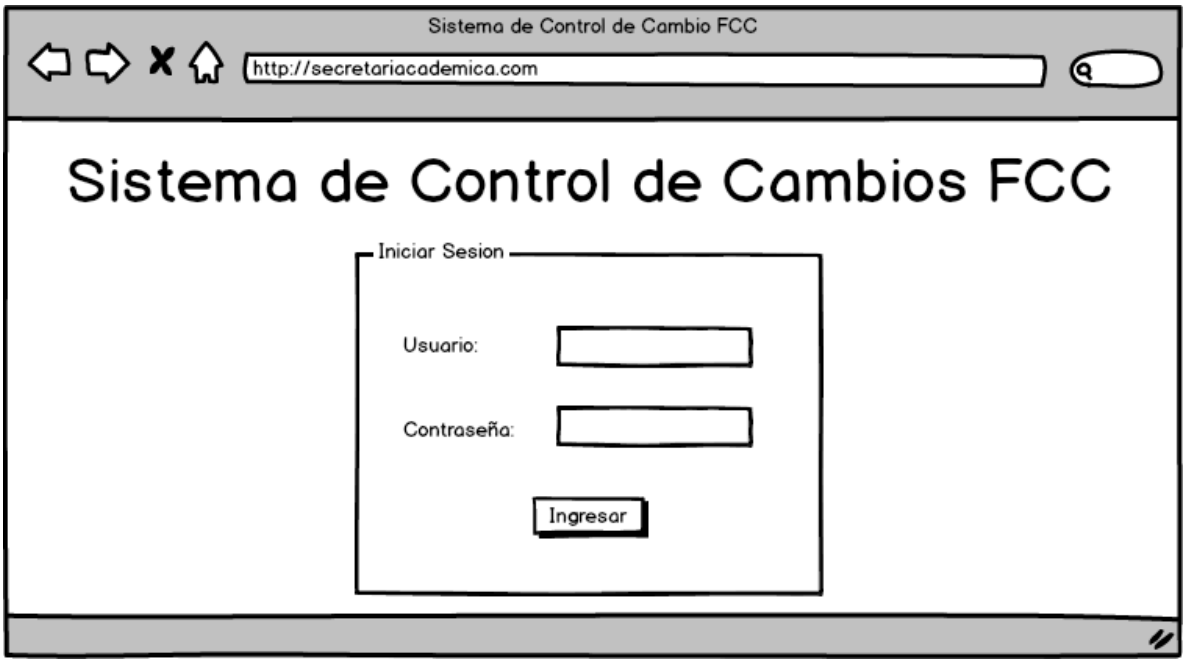


Ilustración 8. Vista - Ingreso al sistema

Alta cambio: cuando el usuario administrador acceda al sistema, podrá visualizar un formulario que deberá llenar para dar de alta una solicitud de cambio que le haya llegado a su correo electrónico.

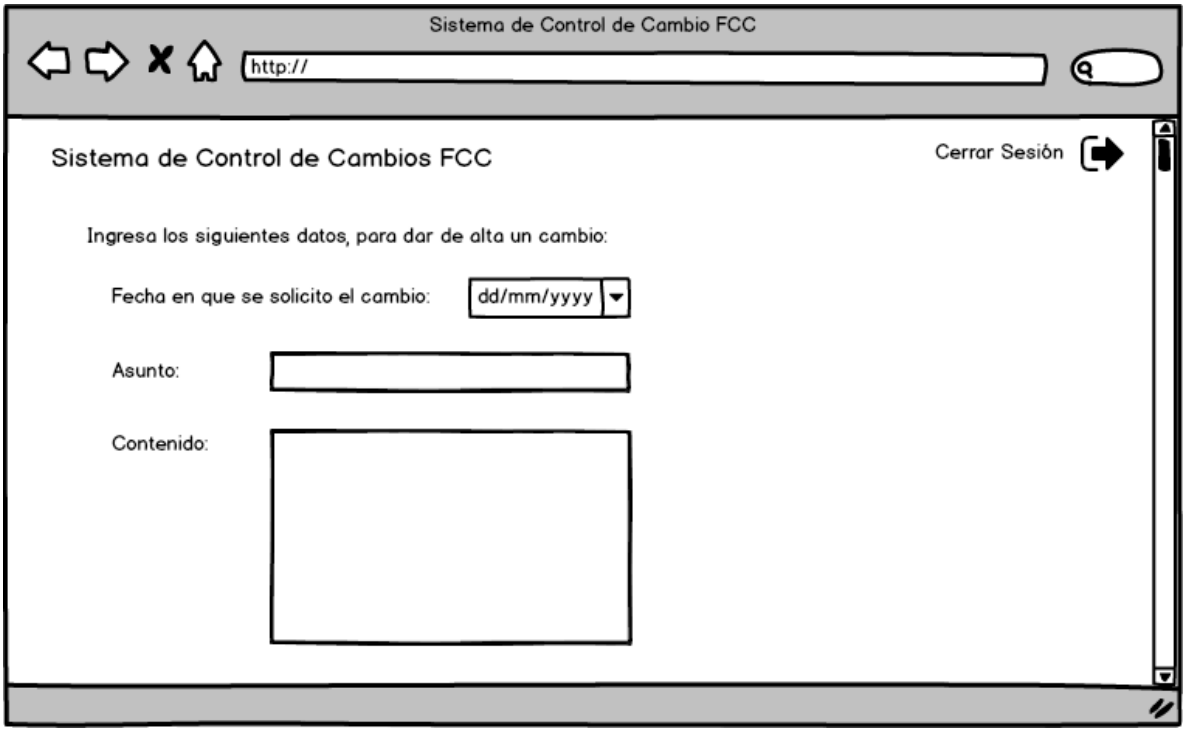


Ilustración 9. Vista - Alta de cambio (1)

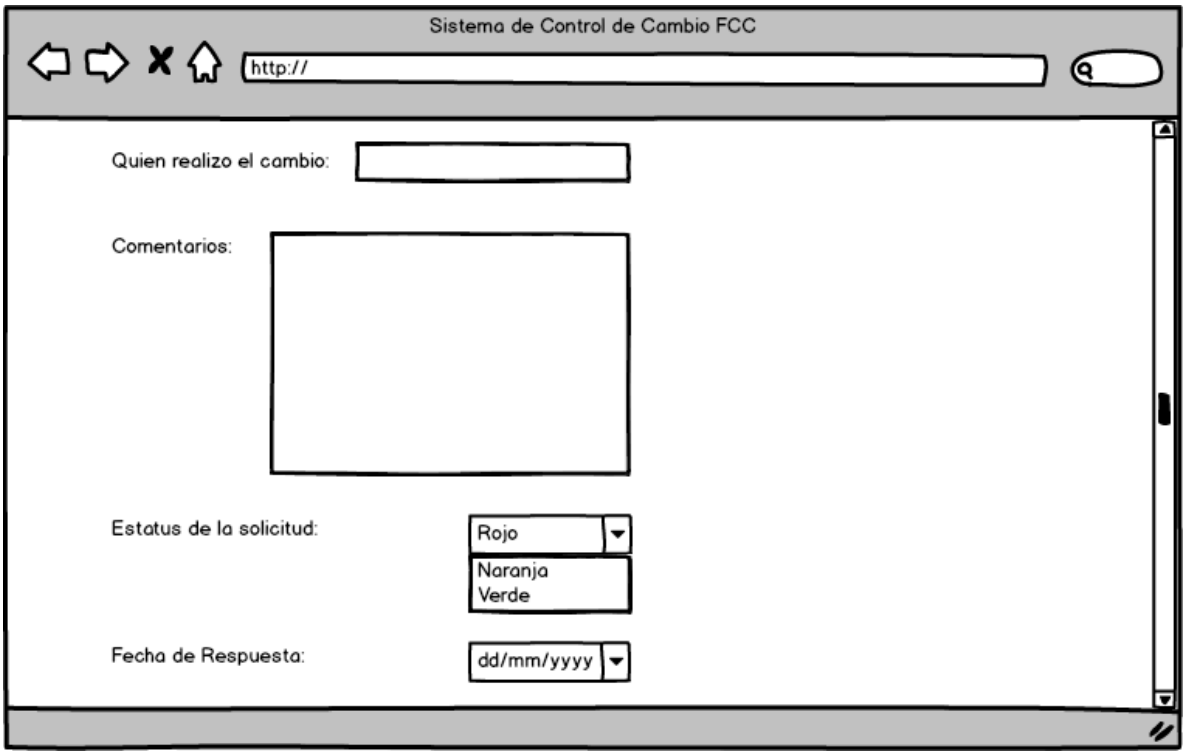


Ilustración 10. Vista - Alta de cambio (2)

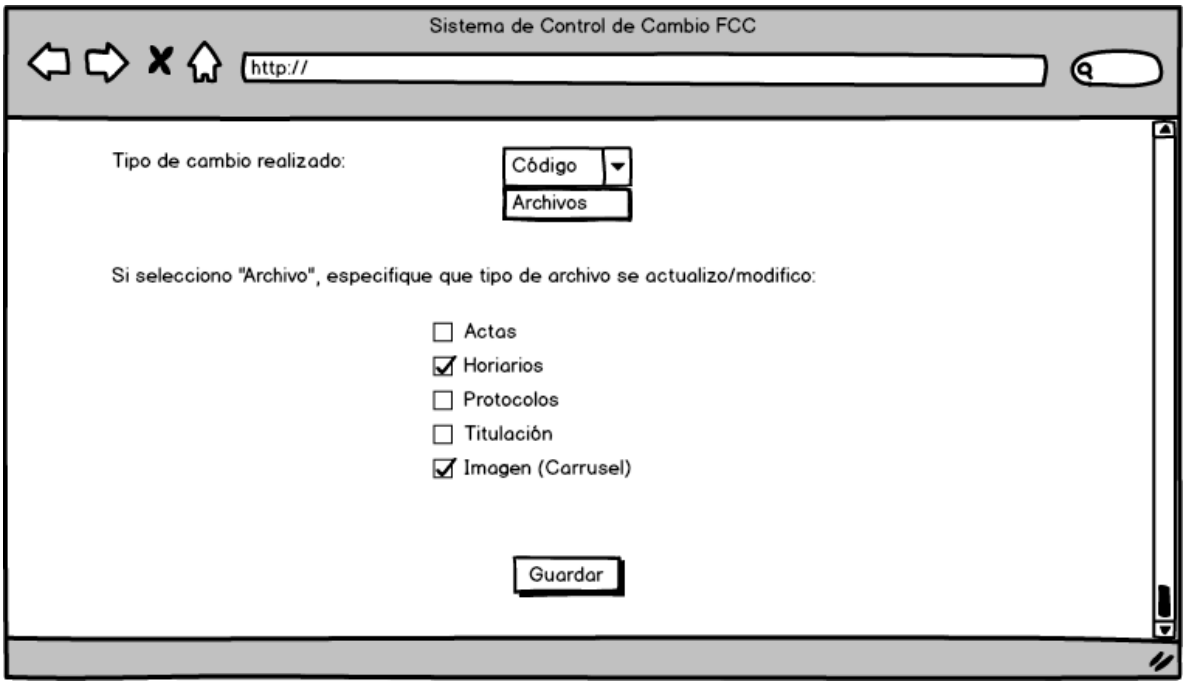


Ilustración 11. Vista - Alta de cambio (3)

Al terminar de llenar el formulario y dar clic en “Guardar”, se mostrar la siguiente vista:

Ilustración 12. Vista - Alta cambio exitoso

Vista root: si el que accede al sistema es el root, se muestra el siguiente menú.

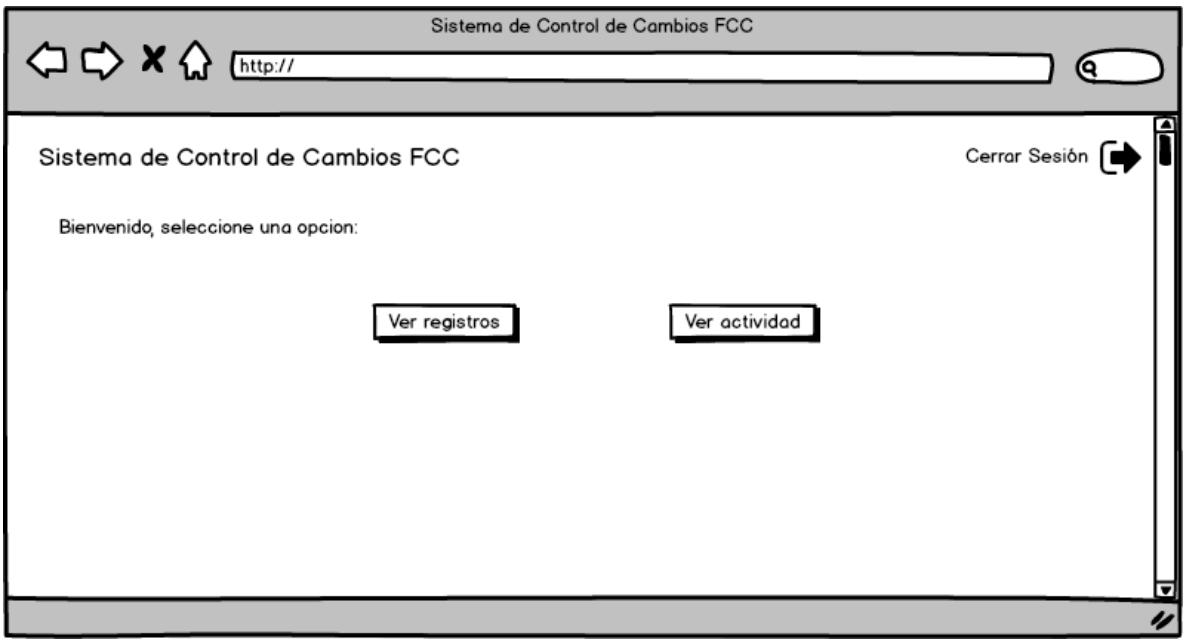


Ilustración 13. Vista - Root

Registro de solicitudes: si root escoge "ver registros", se muestra una tabla con las solicitudes registradas en la base datos, cada solicitud contara con un estatus, que solo el usuario root podrá modificar.

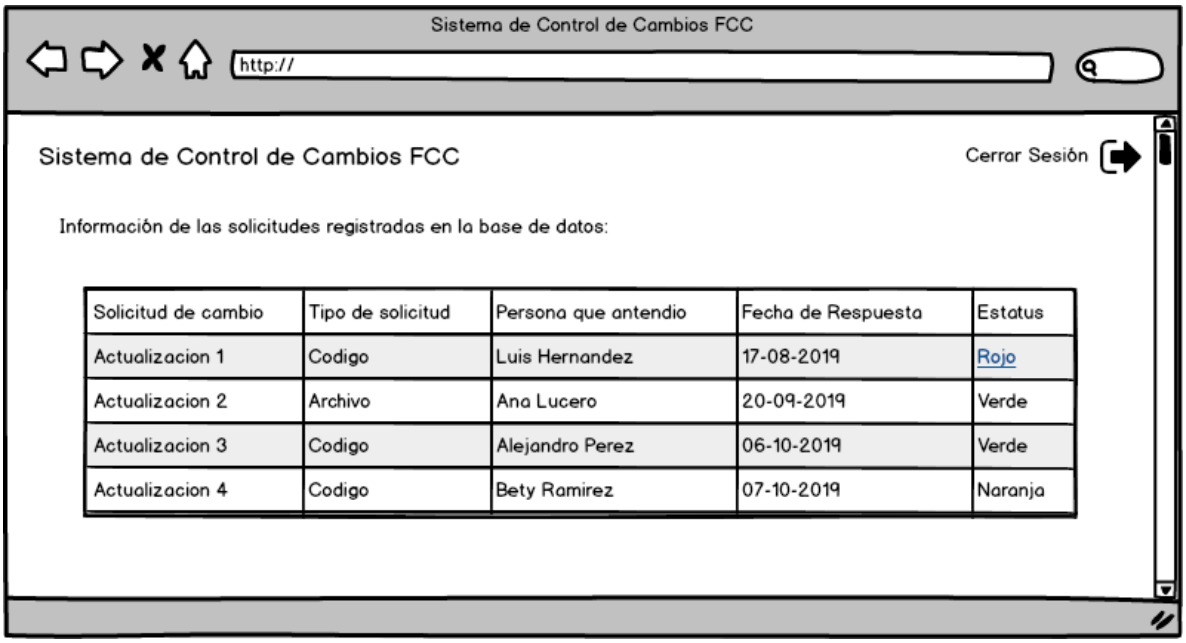


Ilustración 14. Vista - Registro de solicitudes

Modificación de estatus: al seleccionar un estatus de una solicitud, el usuario podrá modificarlo si es que la información esta incorrecta.

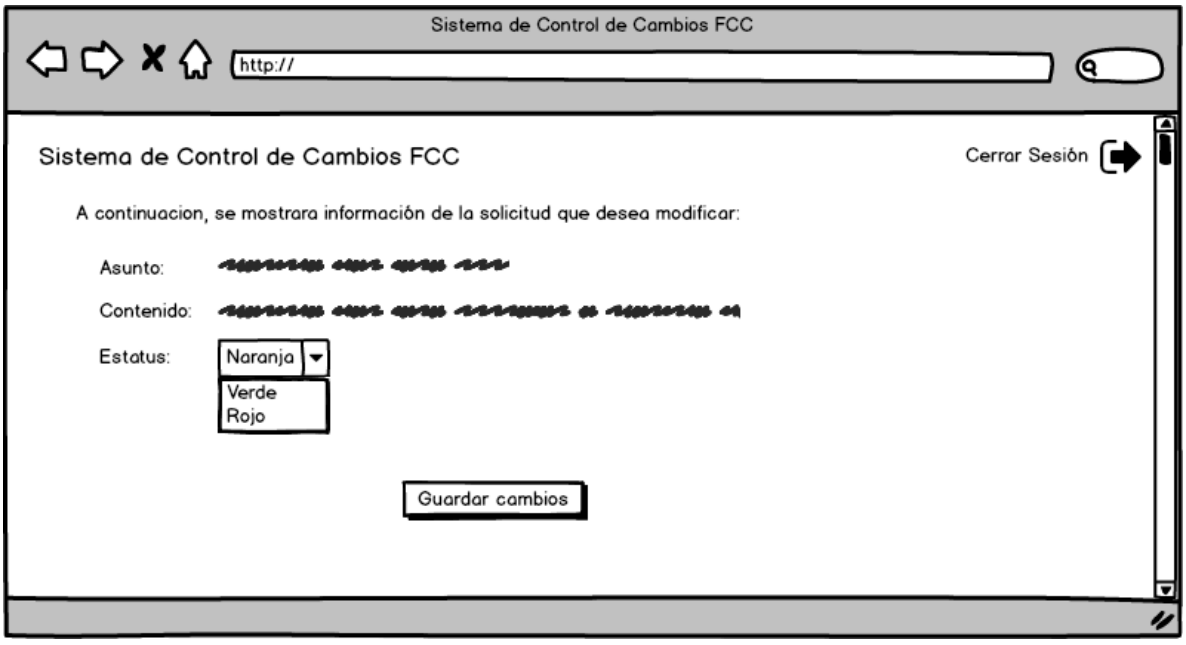


Ilustración 15. Vista- Modificación de estatus

Registro de actividades: al contrario si en la vista “Vista root” selecciona la opción “Ver actividad”, podrá visualizar una tabla con los nombre de los miembros, cuantas actualizaciones sobre el sitio web han realizado y cuáles son las solicitudes que han atendido.

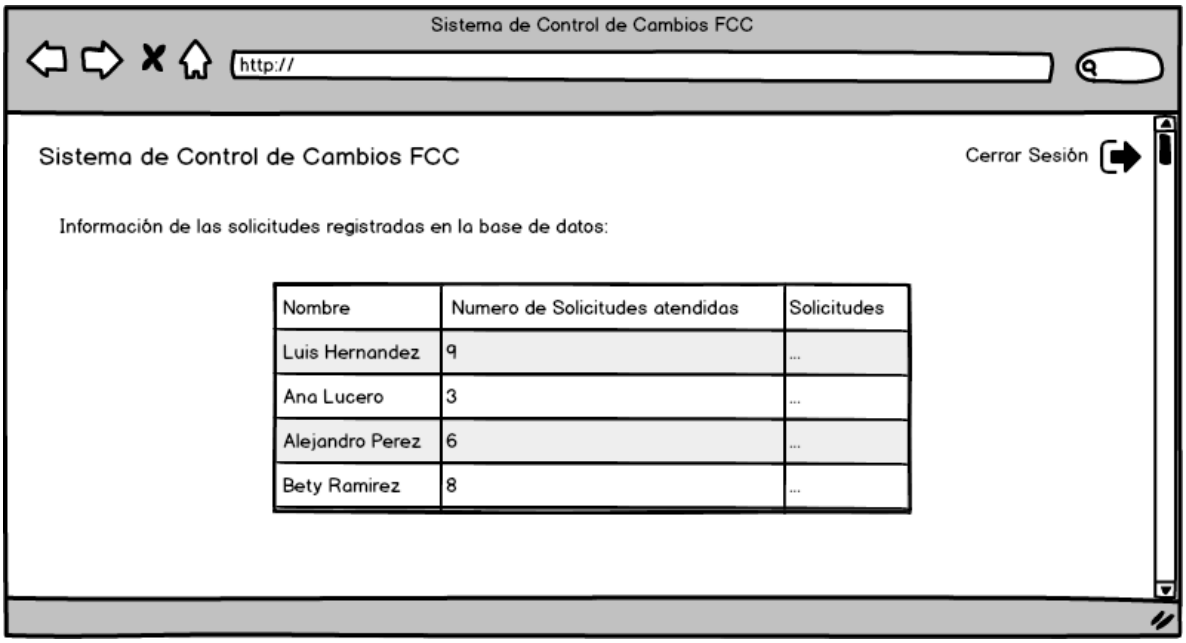


Ilustración 16. Vista - Registro de actividades

Capítulo 6

Pruebas y resultados

Para realizar el desarrollo y pruebas del sistema, se instaló Python para poder crear un entorno virtual y ahí poder crear un proyecto con sus respectivas aplicaciones. Las pruebas realizadas sobre el sistema de control de cambios consistieron en la evaluación del funcionamiento del sistema. Se evaluó que el sistema permitirá el insertar un nuevo registro en la base datos, poder visualizar las solicitudes, editar y/o eliminar dichos registros; también se verifico que cada usuario solo pudiera realizar las tareas que se le asignaron.

6.1 Sistema de Control de Cambios

A continuación se presenta el funcionamiento del sistema de control de cambios. Al iniciar el servidor se muestra la vista de inicio de sesión:

Ilustración 17. Vista - Inicio de Sesión.

Dependiendo de cuál usuario acceda al sistema, se le expondrá información diferente, en el caso de que acceda al sistema el usuario ‘root’ observara lo siguiente:

Ilustración 18. Vista - Muestra de registros

De acuerdo a la anterior imagen, el usuario root podrá visualizar una tabla que contiene todas las solicitudes que se encuentran en la base de datos. De lado derecho en la tabla se encuentran las acciones que puede realizar el usuario root, en este caso editar y/o eliminar una solicitud.

En caso de que seleccione en editar a la primera solicitud, se muestra de la siguiente forma:

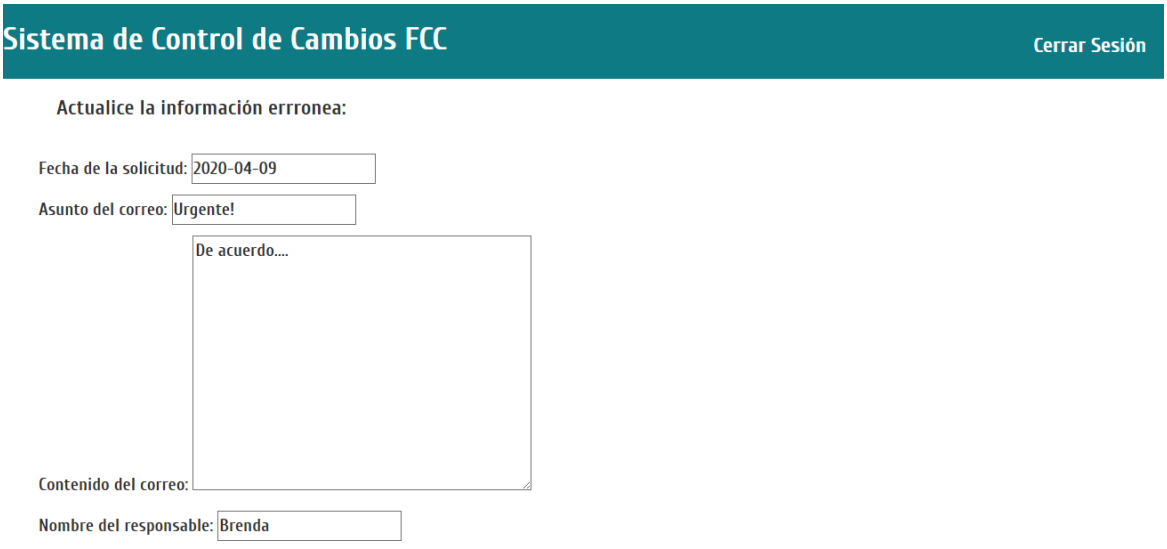


Ilustración 19. Vista - Edición de solicitud.

Se muestra la información de la solicitud que se desea editar. Al terminar de editar la solicitud y dar clic en 'Actualizar', lo regresa a la vista anterior. En contrario si el usuario decidiera eliminar una solicitud le aparecerá la siguiente vista:

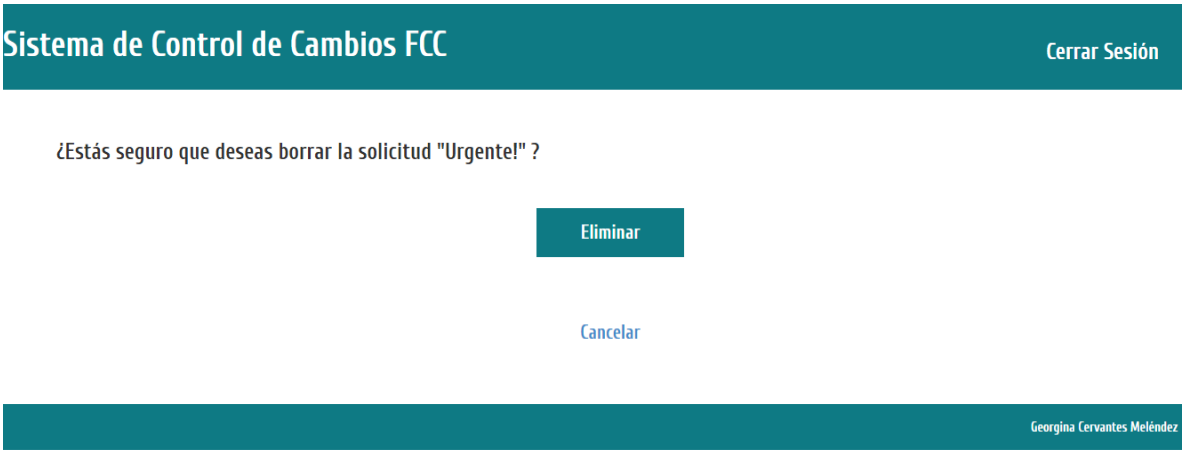


Ilustración 20. Vista - Eliminación de solicitud.

Se presenta la pregunta ¿Estás seguro que deseas borrar la solicitud "solicitud x"? seguido del botón 'Eliminar'. En caso de que se haya equivocado de solicitud, podrá dar clic en 'Cancelar' y lo regresará a la vista anterior.

Aquí terminan las vistas pertenecientes al usuario root, ahora si en el caso que el usuario 'admin' iniciara sesión en el sistema, vera lo siguiente:

Ilustración 21. Vista - Registro de nueva solicitud.

El usuario ‘admin’ vera el mismo formulario que el usuario ‘root’, pero en este caso vacío, para registrar las solicitudes que el atienda mientras está ayudando con el mantenimiento de la página web de la facultad. Al finalizar de rellenar el formulario y dar clic en ‘Guardar’ vera la siguiente vista:

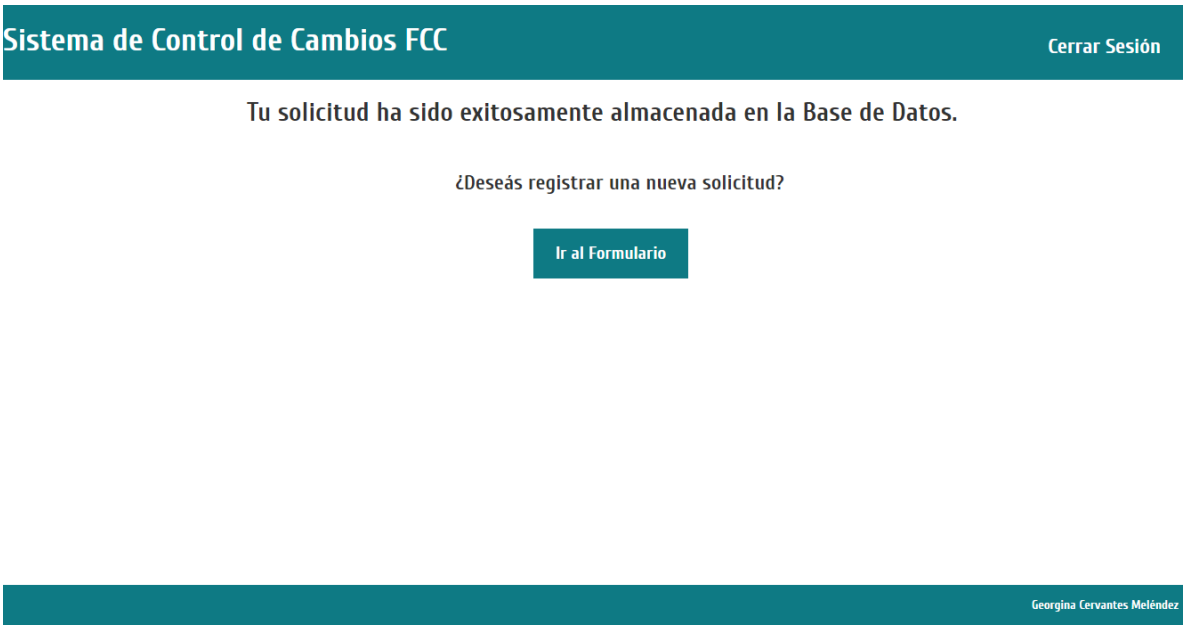


Ilustración 22. Vista - Envió exitoso.

En el caso de que no rellene bien o falte algún campo por llenar le aparecerá un mensaje de advertencia:

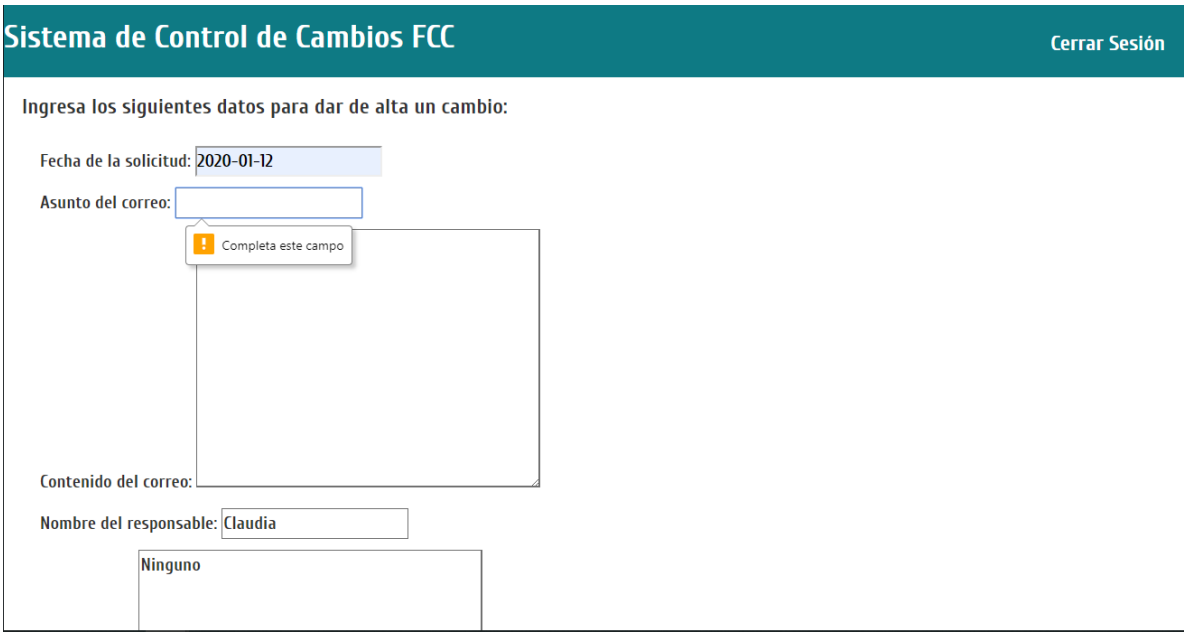


Ilustración 23. Vista - Error uno.

Otro error que se puede producir es si el usuario registra erróneamente la fecha, cuando el formato aceptado es el siguiente (YYYY-MM-DD):

Ilustración 24. Vista - Error dos.

Finalmente cuando el usuario decida salir de sistema, al dar clic en el botón 'Cerrar sesión' este lo dirige al inicio de sesión.

6.2 Administrador de Django.

Por otra parte, Django cuenta con una sección para el administrador donde se agregan los modelos que se definen en el proyecto, también desde el administrador se puede crear a los usuarios que estarán interactuando en la página web. Para tener acceso al administrador se debe crear un súper usuario, con el siguiente comando: *python manage.py createsuperuser.*

Para este proyecto se implementó el siguiente modelo para el formulario que se mostró anteriormente, en este modelo se especifican el nombre de cada campo y su valor.

```python
from django.db import models

class Solicitudes(models.Model):
    fecha_de_la_solicitud = models.DateField()
    asunto_del_correo = models.CharField(max_length=50)
    contenido_del_correo = models.TextField(max_length=500)
    nombre_del_responsable = models.CharField(max_length=50)
    comentarios = models.TextField(max_length=500)
    STATUS = (
        (11, 'Rojo'),
        (12, 'Amarillo'),
        (13, 'Verde'),
        )
    estatus = models.PositiveSmallIntegerField(choices = STATUS,default='R')
    fecha_de_respuesta = models.DateField()
    TYPE = (
        (11, 'Codigo'),
        (12, 'Archivo'),
        )
    tipo_de_cambio_realizado = models.PositiveSmallIntegerField(choices = TYPE)
    FILE = (
        (11,'Actas'),
        (12,'Horarios'),
        (13,'Protocolos'),
        (14,'Titulacion'),
        (15,'Imagen(Carrusel)'),
        (16,'Ninguno'),
        )
    especifique_archivo = models.PositiveSmallIntegerField(choices = FILE)

    #MUESTRA TITULO DE CADA REGISTRO EN LA BD
    def __str__(self):
        return self.asunto_del_correo
```

Ilustración 25. Django - Modelo Formulario.

Para poder visualizar nuestro modelo en el administrador de Django debemos ir a consola y correr el siguiente comando: '*python migrate.py migrate',* en el administrador aparecerá así:

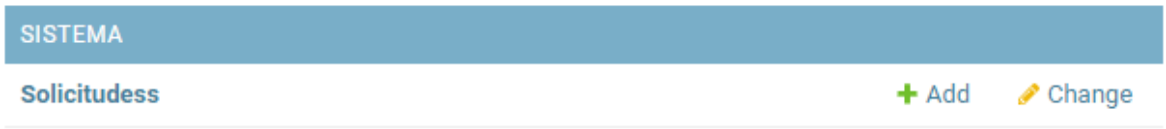
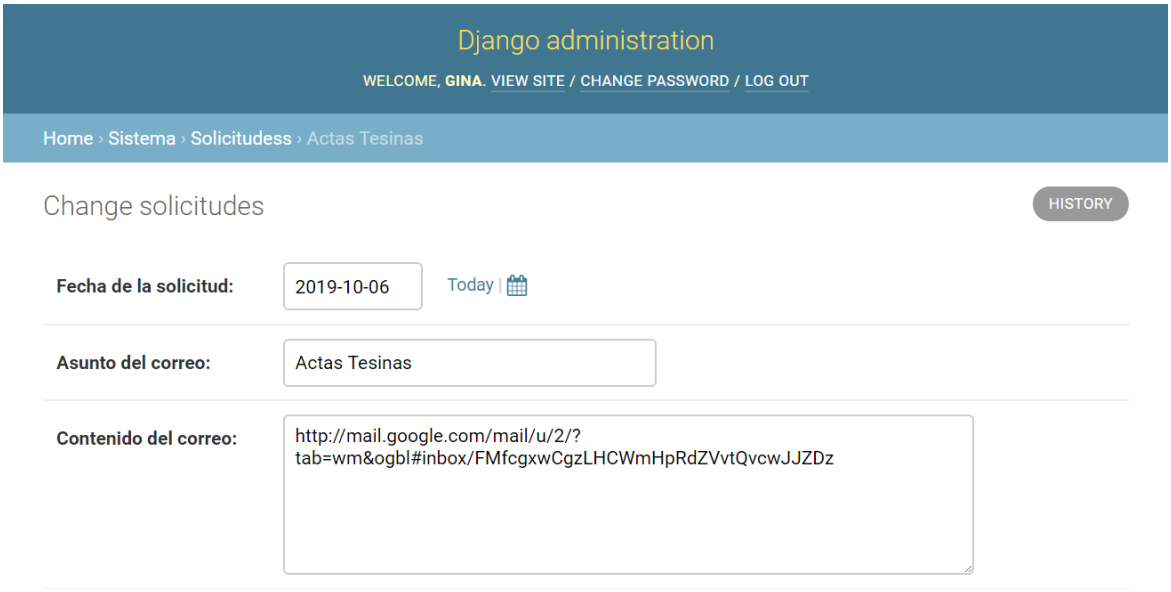


Ilustración 26. Administrador - Modelo.

Desde aquí también podemos ver los registros que se han agregado a la base de datos, solo damos clic en el modelo que creamos:

Ilustración 27. Administrador - Registros.

Desde aquí también podemos agregar, editar y/o eliminar una solicitud. A continuación se muestran unos ejemplos de cómo se ven los registros en base de datos:



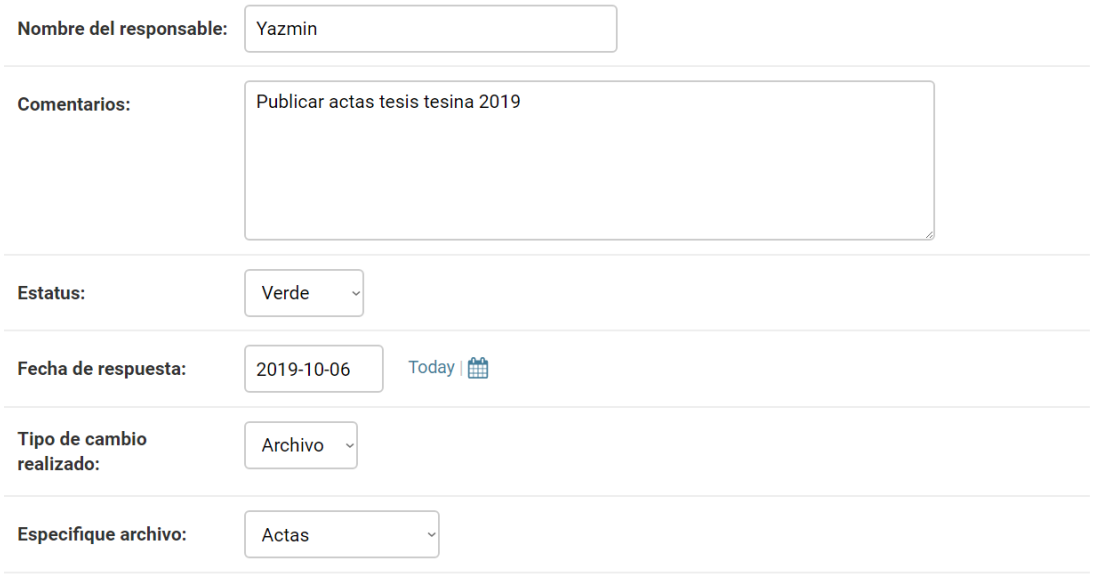
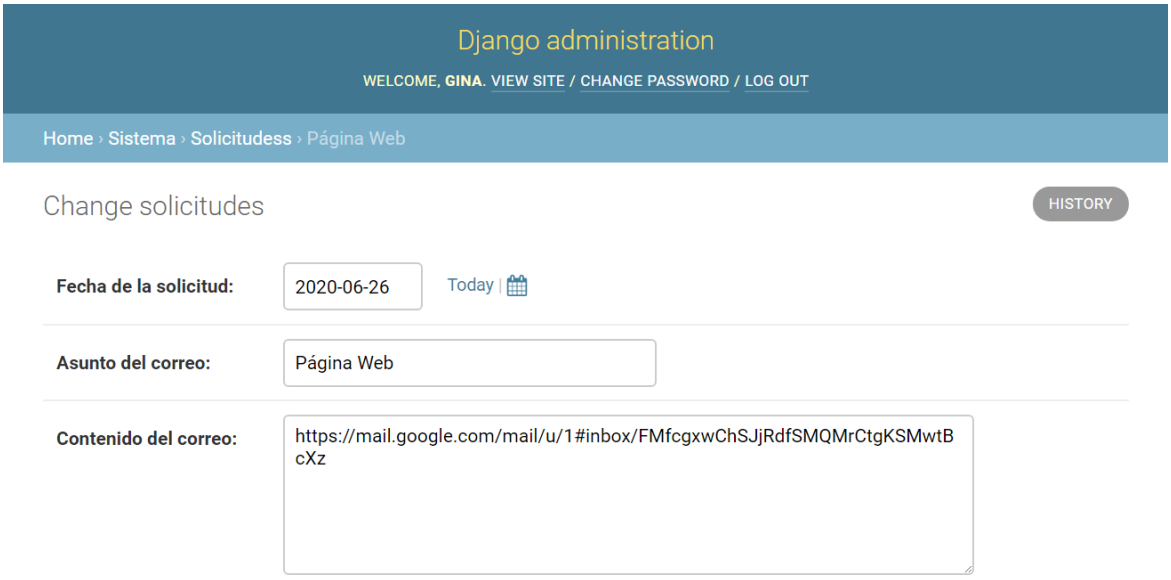


Ilustración 28. Administrador- Ejemplo 1.



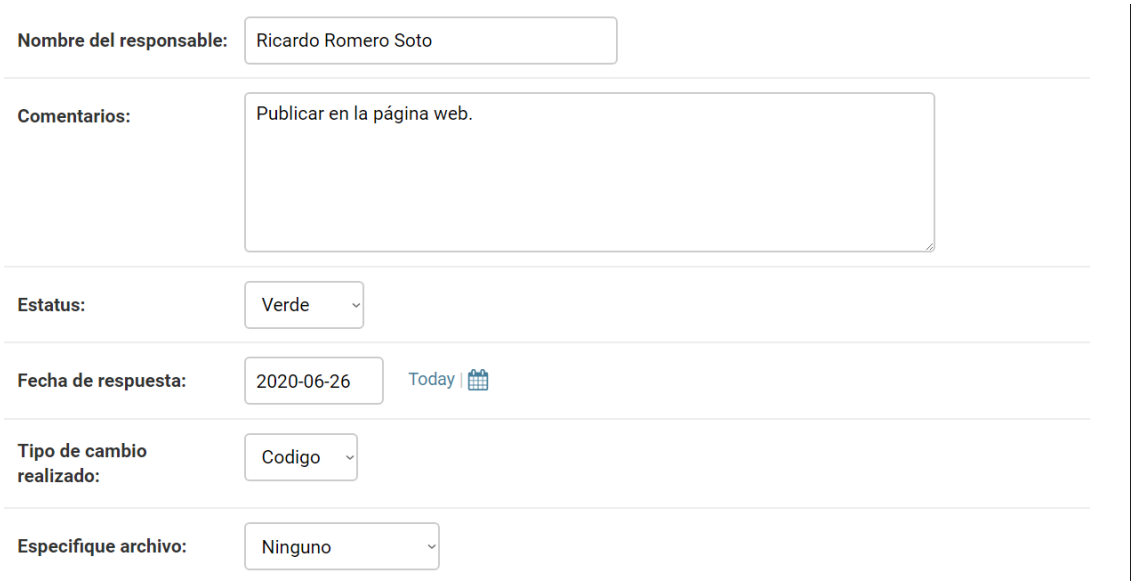
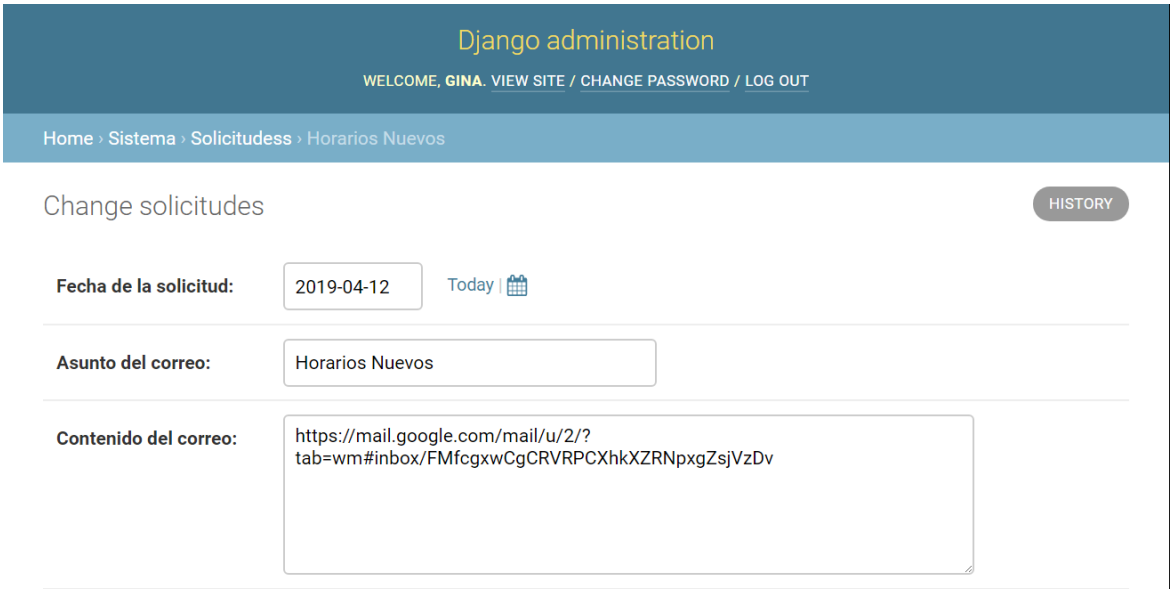


Ilustración 29. Administrador - Ejemplo 2.



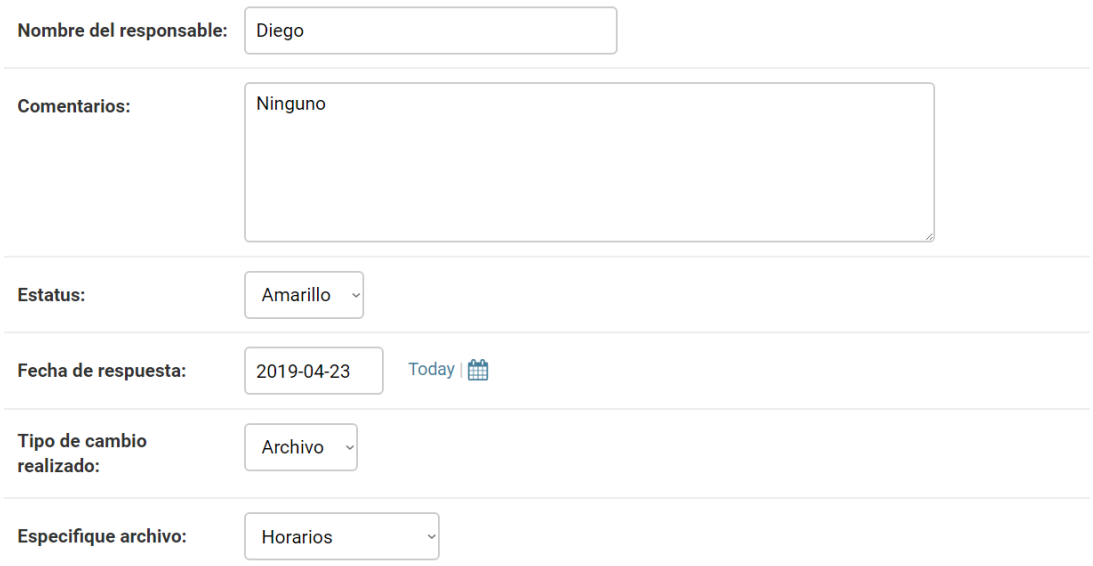


Ilustración 30. Administrador - Ejemplo 3.

Por otra parte el administrador de Django nos permite crear a los usuarios que interactúan en nuestra página web. Esto lo podemos encontrar en la sección de autentificación y autorización, ahí podemos tanto crear usuarios como grupos a los que los usuarios estarán afiliados

Ilustración 31. Administrador - Autentificación y Autorización.

Para el proyecto se contemplaron dos tipos de usuarios, uno que es parte del staff y otro que no, durante el desarrollo se crearon tres usuarios, el usuario '*gina*' se utilizó para la creación de los otros usuarios y asignarles permisos.

Ilustración 32. Administrador - Usuarios.

En conclusión, solo el súper usuario puede tener acceso a esta sección de Django, él puede crear nuevos modelos, agregar editar y eliminar registros. Crear nuevos usuarios y asignarles permisos y también crear grupos a los que los usuarios estarán ligados.

Conclusiones

El objetivo fundamental de esta tesis es abordar el problema de guardar los datos de las solicitudes de mantenimiento hacia la página web de la facultad que llegan al laboratorio ASR, clave para la identificación de los cambios realizados en la página y quien los hizo.

El control de cambios es uno de los mayores problemas a afrontar en los proyectos de software. Para solventar este problema se recurre a las herramientas de control de cambios que facilitan el almacenamiento de los elementos a gestionar la recuperación de cada uno de ellos y el registro histórico e identificación de cada una de las modificaciones realizadas en los sucesivos cambios realizados del código y documentación del proyecto.

Se ha presentado un estudio de las principales herramientas para realizar el control de versiones en proyecto de software, analizando tanto sus características como sus funcionalidades y carencias con el fin de comprobar cuáles serían las funcionalidades esenciales para la creación de una herramienta idónea que satisfaga las necesidades reales de un equipo de proyecto.

Así pues, la principal aportación de este trabajo consiste en la aportación del diseño e implementación de un sistema de control de cambios en el que se trata de satisfacer las necesidades de los usuarios, tanto los estudiantes responsables del mantenimiento de la página web de la facultad, como los profesores a cargo del laboratorio.

Existen diversas herramientas que brindan soporte al usuario desde un servidor web con la ayuda de apache, Python, el framework y las dependencias que se utilizan y como en conjunto finalmente permiten que exista el sistema como una solución.

El sistema de control de cambios de la FCC, busca ayudar al estudiante a registrar las solicitudes que ha atendido en su respectivo turno y al profesor a tener un registro de que estudiantes han estado trabajando en el mantenimiento de la página y cuántas solicitudes han sido atendidas correctamente. Dicho sistema también permitiría encontrar los cambios que se han realizado tanto en el código, como en los archivos almacenados dentro del proyecto, lo que ayuda mucho en el proceso de la depuración de un proyecto.

Trabajo Futuro

Como continuación de este trabajo de tesis y como en cualquier otro proyecto de investigación, existen diversas líneas de investigación que quedan abiertas y en las que es posible continuar trabajando. Durante el desarrollo de esta tesis han surgido algunas líneas futuras sé que han dejado abiertas y se esperan desarrollar en el futuro.

A continuación se presentan algunos trabajos futuros que pueden desarrollarse como resultado de esta investigación o que, por exceder el alcance de esta tesis, no han podido ser tratados con la suficiente profundidad. Además se sugiere algunos desarrollos específicos para apodar y mejorar lo propuesto en la tesis.

Entre los posibles trabajos futuros se destacan:

- Es necesario señalar que el diseño que se propuso no es el más óptimo y esto deja la posibilidad de darle un seguimiento a este trabajo.
- Implementar una opción donde se pueda guardar un respaldo de la página web de antes y después de realizar los cambios solicitados.
- Alojar el sistema control de cambio con una IP universitaria, así cada vez que se quiera acceder al sistema, lo puedan hacer a través de su propia computadora mientras se encuentren en la universidad.
- Extender las funciones que puede realizar el profesor a través del sistema, por ejemplo realizar anuncios o recordatorios a los responsables del mantenimiento, etc.
- Convertir la página web en una aplicación móvil, no será necesario realizar el registro de las solicitudes desde una laptop o computadora de escritorio, también así el profesor responsable podrá tener un seguimiento más cercano a la actividad que realizan sus estudiantes a cargo del mantenimiento.

Bibliografía

1. Accurev Inc. (2010). AccuRev Concepts Manual.
2. Bazaar Developers. (2011). Bazaar Tutorial.
3. Braude, E. and Polo Usaola, M. (2003). Ingeniería de software. Madrid: Ra-Ma.
4. Cederqvist, P., & Pesch, R. (1993). Version management with CVS. Signum Support AB.
5. ChangeGear vs. Freshservice vs. Whatfix vs. ServiceNow Comparison. (2020, 24 junio). Recuperado 29 de junio de 2020, de https://sourceforge.net/software/compare/ChangeGear-vs-Freshservice-vs-Whatfix-vs-ServiceNow/
6. Chacon, S., & Straub, B. (2014). Git - Acerca del Control de Versiones. Recuperado 22 de agosto de 2020, de https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones
7. David, M. (2010). HTML5. Oxford: Focal.
8. Django, D. (s.f.). Django documentation | Django documentation | Django. Recuperado 19 enero, 2020, de https://docs.djangoproject.com/en/3.0/
9. EcuRed. (s.f.). SQLite - EcuRed. Recuperado 27 octubre, 2019, de https://www.ecured.cu/SQLite
10. García Pérez, E. I. G. P. Eduardo Ismael. (s.f.). Que es Git? Recuperado 3 octubre, 2019, de https://codigofacilito.com/articulos/que-es-git
11. Galfullin, B. N., & Novichkov, A. N. (2000). ClearCase-source code managing system. Microwave Conference, 2000. Microwave and Telecommunication Technology. 2000 10th International Crimean, 90-93.
12. Gustavo, B. (2019, 17 julio). ¿Qué es Apache? Descripción completa del servidor web Apache. Recuperado 27 octubre, 2019, de https://www.hostinger.mx/tutoriales/que-es-apache/
13. Manuel Rubio, M. R. (2010, 16 noviembre). ¿Cómo funciona el sistema web? Recuperado 26 octubre, 2019, de https://altenwald.org/2010/11/16/como-funciona-el-sistema-web/
14. Patricia Miravet, P. M., Carlos Alba, C. A., Vicente Rodríguez, V. R., & Luis Fernández, L. F. (2011, 8 julio). Análisis y evaluación de herramientas de control de

versiones en proyectos software. Recuperado 16 abril, 2019, de https://www.aeipro.com/files/congresos/2011huesca/CIIP11_2390_2405.3423.pdf

15. Pressman, R. (1988). Ingeniería del software. Madrid: McGraw-Hill.
16. ¿Por qué usar Django?, J. C. M. U. Jean Carlos Mariños Urquiaga. (2016, 20 junio). ¿Por qué usar Django? Recuperado 3 octubre, 2019, de https://devcode.la/blog/por-que-usar-django/
17. Qué es SCRUM. (2018, 9 octubre). Recuperado 15 Abril, 2019, de https://proyectosagiles.org/que-es-scrum/
18. Russell Smith, R. S. (2017, 22 noviembre). 3 Reasons Why Change Control Is Important - Lepide Blog: A Guide to IT Security, Compliance and IT Operations. Recuperado 26 octubre, 2019, de https://www.lepide.com/blog/three-reasons-why-change-control-is-important/
19. Schimpf, J. (2010). Fossil Version Control A Users Guide.
20. Spartasystems.com. (2019). Change Control Management Software | Sparta Systems. Recuperado 25 Abril, 2019, de https://www.spartasystems.com/solutions/change-control
21. Tom Alby, T. A. (2010, 26 agosto). Change Control System [Tool]. Recuperado 15 Abril, 2019, de https://project-management-knowledge.com/definitions/c/change-control-system-tool/
22. Top 10 Change Management Software Solutions in 2020. (2020, 4 junio). Recuperado 24 de junio de 2020, de https://www.softwaretestinghelp.com/change-management-software/

Anexos

Anexo 1

Esta sección ofrece información adicional importante al momento de la instalación, desarrollo y pruebas del sistema de control de cambios.

Instalación

Para la implementación de la propuesta hecha en esta tesis, se debe tener lo siguiente instalado:

- Python.
- Framework Django.
- Editor de Texto, en este caso Sublime Text.

Primero se instaló Python, y desde consola se confirma que la instalación fue correcta y verificamos la versión que se instaló:

```
C:\Users\julio>python --version
Python 3.8.1

C:\Users\julio>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Ilustración 33. Consola 1.

Se creó la carpeta que almacenaría el proyecto, y desde ahí se crearía el entorno virtual para crear nuestro proyecto en Django:

```
C:\Users\julio\proyecto>python -m venv sistema
```

Ilustración 34. Consola 2.

Activamos el entorno virtual, al activarlo aparecerá el nombre del entorno al inicio de línea de comando:

```
C:\Users\julio\proyecto>sistema\Scripts\Activate

(sistema) C:\Users\julio\proyecto>
```

Ilustración 35. Consola 3.

Para instalar Django, antes se debe verificar que se tiene el comando pip actualizado, si no es así, se debe actualizar con el siguiente comando:

```
(sistema) C:\Users\julio\proyecto>python -m pip install --upgrade pip
Collecting pip
  Using cached https://files.pythonhosted.org/packages/00/b6/9cfa56b4081ad13874b0c6
f96af8ce16cfbc1cb06bedf8e9164ce5551ec1/pip-19.3.1-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 19.2.3
    Uninstalling pip-19.2.3:
      Successfully uninstalled pip-19.2.3
Successfully installed pip-19.3.1
```

Ilustración 36. Consola 4.

Y finalmente podremos instalar Django con el siguiente comando:

```
(sistema) C:\Users\julio\proyecto>pip install django
Collecting django
  Using cached https://files.pythonhosted.org/packages/55/d1/8ade70e65fa157e1903fe4
078305ca53b6819ab212d9fbbe5755afc8ea2e/Django-3.0.2-py3-none-any.whl
Collecting sqlparse>=0.2.2
  Using cached https://files.pythonhosted.org/packages/ef/53/900f7d2a54557c6a378865
85a91336520e5539e3ae2423ff1102daf4f3a7/sqlparse-0.3.0-py2.py3-none-any.whl
Collecting asgiref~=3.2
  Using cached https://files.pythonhosted.org/packages/a5/cb/5a235b605a9753ebcb2730
c75e610fb51c8cab3f01230080a8229fa36adb/asgiref-3.2.3-py2.py3-none-any.whl
Collecting pytz
  Using cached https://files.pythonhosted.org/packages/e7/f9/f0b53f88060247251bf481
fa6ea62cd0d25bf1b11a87888e53ce5b7c8ad2/pytz-2019.3-py2.py3-none-any.whl
Installing collected packages: sqlparse, asgiref, pytz, django
Successfully installed asgiref-3.2.3 django-3.0.2 pytz-2019.3 sqlparse-0.3.0
```

Ilustración 37. Consola 5.

Después de instalar Django, se creó que el proyecto que contendrá las aplicaciones del sistema:

```
(sistema) C:\Users\julio\proyecto>django-admin.exe startproject control
```

Ilustración 38. Consola 6.

Para poder iniciar el servidor, necesitamos movernos a la carpeta ‘control’ y ahí se deberá encontrar el archivo manage.py, eso no ayudara a iniciarlo:

```
(sistema) C:\Users\julio\proyecto>cd control

(sistema) C:\Users\julio\proyecto\control>dir
 El volumen de la unidad C es Acer
 El número de serie del volumen es: EA5A-0141

 Directorio de C:\Users\julio\proyecto\control

20/01/2020  05:38 p. m.    <DIR>          .
20/01/2020  05:38 p. m.    <DIR>          ..
20/01/2020  05:38 p. m.    <DIR>          control
20/01/2020  05:38 p. m.               648 manage.py
               1 archivos            648 bytes
               3 dirs  593,751,277,568 bytes libres

(sistema) C:\Users\julio\proyecto\control>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you ap
ply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 20, 2020 - 17:41:38
Django version 3.0.2, using settings 'control.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Ilustración 39. Consola 7.

En la anterior imagen, al iniciar el servidor te muestra la dirección url a la se debe acceder para verificar si se instaló de forma correcta Django, en este caso la dirección es: http://127.0.0.1:8000/ , se vera de la siguiente forma:

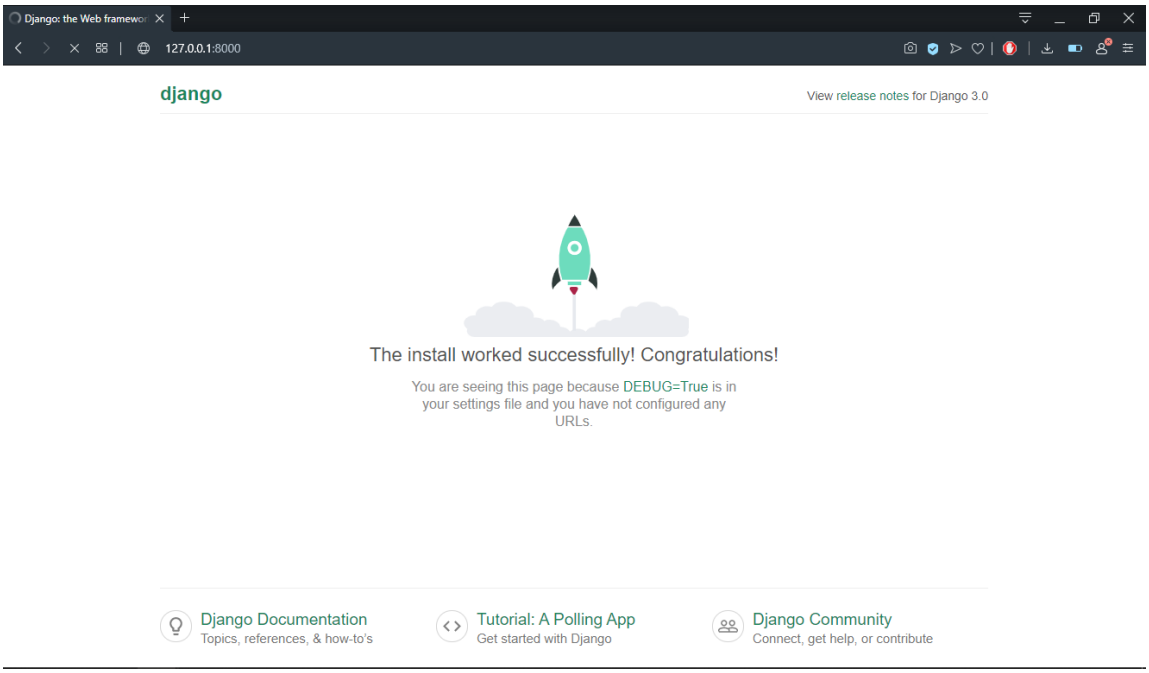


Ilustración 40. Navegador – Primera prueba.

Por último, para comenzar a desarrollar la interfaz del sistema se creó una aplicación:

```
(sistema) C:\Users\julio\proyecto\control>django-admin.exe  startapp version
```

Ilustración 41. Consola 8.

'Versión' es el nombre de la aplicación, este nombre puede ser cualquiera.

En mi caso, la carpeta que contiene mi proyecto quedo de la siguiente forma, donde se agregaron los templates de cada vista en este caso, formulario con: registro y envió del formulario, registro solamente con el inicio de sesión, y root con: visualización, edición y eliminación de solicitudes ya registradas en las base de datos:

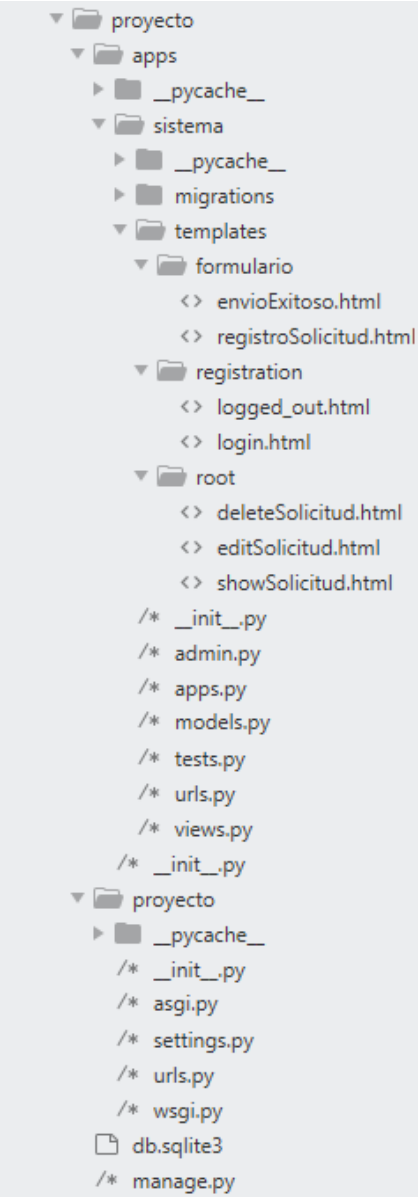


Ilustración 42. Estructura del proyecto.

Ejemplo primera vista en Django

Ahora, como ejemplo para poder ver nuestra primera vista, es decir, el inicio de sesión primero debemos realizar ciertas configuraciones en el archivo '*proyecto/proyecto/**settings.py***'. Primero debemos escribir nuestra aplicación en la sección de '*INSTALLED_APPS*' de la siguiente forma:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    #APLICACIONES
    'apps.sistema',
]
```

Ilustración 43. Archivo 'settings.py' – Installed_apps.

También debemos especificar el nombre de la carpeta donde estarán las vistas de nuestro proyecto, en este caso están en la carpeta '*templates*':

```
TEMPATES_DIRS = (
    RUTA_PROYECTO.child('templates'),
    )
```

Ilustración 44. Archivo 'settings.py' - Templates_dirs.

Aquí terminan las configuraciones en el archivo '*proyecto/proyecto/**settings.py***', ahora pasamos a la carpeta '*templates*', donde estarán nuestra vistas. Pasamos a modificar el archivo '*proyecto/proyecto/**urls.py***', donde especificamos como queremos que aparezca la dirección en nuestra navegador y a que urls se dirigirá. La dirección se vera de la siguiente forma http://127.0.0.1:8000/sistema/ .

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('sistema/', include('apps.sistema.urls')),
]
```

Ilustración 45. Archivo 'urls.py' - urlpatterns.

En la carpeta '*registration*' creamos un archivo llamado '*proyecto/apps/sistema/templates/registration/**login.html***' el cual dentro de la etiqueta <*body*> contiene lo siguiente:

```
<nav>
    <div>
        <ul>
            <li><h1>Sistema de Control de Cambios FCC</h1></li>
        </ul>
    </div>
</nav>

{% if form.errors%}
<p> Tu usuario y contraseña no coinciden. Intentalo de nuevo </p>
{% endif %}

{% if next %}
    {% if user.is_authenticated %}
    <p> Tu cuenta no tiene acceso a esta página.</p>
    {% else %}
    <p> Ingresa para ver esta página. </p>
    {% endif %}
{% endif%}

<form method="POST" action="{% url 'login' %}">
    {% csrf_token %}

    <center>
        <h2>Bienvenido!</h2><br>
        <legend><h2>Inicia Sesión</h2></legend>

        <div class="page-form">
            <td>{{ form.username.label_tag }}</td>
            <td>{{ form.username }}</td>
        </div>
        <br>
        <div class="page-form">
            <td>{{ form.password.label_tag }}</td>
            <td>{{ form.password }}</td>
        </div>
        <br>
        <div class="page-form">
            <input type="submit" value="Ingresar">
            <input type="hidden" name="next" value="{{ next }}">
        </div>
        <br>
    </center>

</form>
```

Ilustración 46. Archivo 'login.html'.

Contiene un *form* que evaluará los datos que el usuario ingrese en las cajas de texto, con la acción '*{% url 'login' %}*'.

Es importante también poner en cada vista que contenga formulario que utilizan el método POST deben incluir el token CSRF en la plantilla. La etiqueta de plantilla *{% csrf_token %}* generará un campo oculto y garantizará que la cookie esté configurada en la respuesta.

```
<form method="POST" enctype="multipart-form-data">
    {% csrf_token %}
    {{form.as_p}}
    <br>
    <center>
        <input type="submit" value="Guardar">
    </center>
</form>
```

Ilustración 47. Ejemplo CSFR_Token

Ahora modificaremos el archivo '*proyecto/apps/sistema/**views.py***', en donde definimos cada vista que tenga nuestra página web. En este caso, como tenemos dos usuarios diferentes, queremos que a los dos nos muestre dos diferentes vistas. Entonces tenemos que si el usuario esta autenticado y es parte de staff lo redirigirás a la página A, en caso contrario lo redirigirás a la página B.

```
def ValidarUsuario(request):
    if request.user.is_authenticated:
        if request.user.is_staff:
            return redirect('/sistema/show/')
        else:
            return redirect('/sistema/formulario/')
        #elif request.user.is_superuser:
        #   return redirect('/admin/')
    return redirect('login')
```

Ilustración 48. Archivo 'views.py' – Validar usuario.

Después de definir nuestra vista, pasamos al archivo '*proyecto/apps/sistema/**urls.py***', donde definimos la dirección a la que el usuario tendrá acceso al inicio de sesión en el navegador. En las primeras comillas podemos escribir cualquier palabra clave que al escribirlo en el navegador nos muestra la vista de inicio de sesión, por ejemplo http://127.0.0.1:8000/sistema/accounts/**iniciosesion/** , en este caso deje solo las comillas entonces en el url se vera de la siguiente forma http://127.0.0.1:8000/sistema/accounts/**login/** que es el predeterminado, y finalmente hacemos referencia a la vista y le asignamos un nombre la dirección.

```
urlpatterns = [
    #VALIDACION USUARIO
    path('accounts/', include('django.contrib.auth.urls')),
    path('', views.ValidarUsuario, name='validar_usuario'),  ]
```

Ilustración 49. Archivo 'urls.py' – Nueva dirección.

Finalmente regresamos al archivo '*proyecto/proyecto/**settings.py***' donde definimos la variable 'LOGIN_REDIRECT_URL' a la cual le asignamos el nombre de la dirección de la vista de inicio de sesión '*validar_usuario'*, la cual definimos en el archivo '*proyecto/apps/sistema/**urls.py***'.

```
from django.urls import reverse_lazy
LOGIN_REDIRECT_URL = reverse_lazy('validar_usuario')
```

Ilustración 50. Archivo 'settings.py' - Login_Redirect_URL.

Por supuesto después de agregarle estilos a tu página web, vamos a la consola e iniciamos el servidor y vamos a la siguiente dirección: http://127.0.0.1:8000/sistema/accounts/login/, y el resultado es el siguiente:

Ilustración 51. Navegador - Primera vista ejemplo.