



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“PJCAT: BÚSQUEDA DE ARCHIVOS CON SOPORTE MULTI-
DISPOSITIVOS”

INFORME DE PROYECTO INTEGRADOR

Previa a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES

LEONARDO EDWIN TAMAYO DELGADO

MARIO FERNANDO IZQUIERDO CHAVARRÍA

GUAYAQUIL – ECUADOR

AÑO: 2016

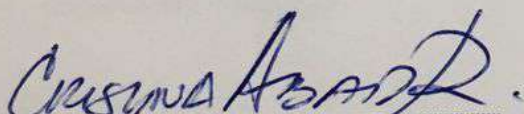
AGRADECIMIENTO

Nuestros más sinceros agradecimientos a Dios, a nuestras familias que siempre nos ayudaron cuando lo necesitamos, a nuestros maestros que fueron piezas clave en nuestra formación como profesionales, a nuestros compañeros de trabajo en Wishbox por el apoyo en todo momento, y a la Dra. Cristina Abad que fue nuestra profesora y guía durante el proceso de graduación, gracias a quien pudimos sacar adelante este proyecto.

DEDICATORIA

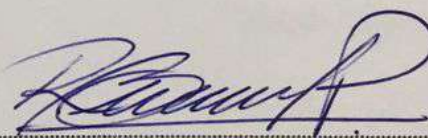
El presente proyecto lo dedicamos a nuestras familias que estuvieron siempre ahí para apoyarnos en cada paso que dimos a lo largo de nuestros estudios, permitiéndonos superarnos siempre.

TRIBUNAL DE EVALUACIÓN



Ph.D. Cristina Abad Robalino

PROFESOR EVALUADOR



M.Sc. Rafael Ignacio Bonilla Armijos

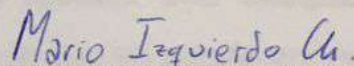
PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Leonardo Tamayo



Mario Izquierdo

RESUMEN

Un problema muy común hoy en día es que al tener muchos computadores, muchos dispositivos de almacenamiento extraíbles (como memorias USB o discos duros externos), y hasta cuentas de almacenamiento en la nube como Dropbox, se vuelve muy sencillo perder noción de dónde se ha guardado tal o cual documento. Esto se vuelve especialmente cierto para personas poco organizadas que, con tener un solo computador sufren de este problema. Es por eso que nace la iniciativa de PJCat, en la que se busca catalogar todos los archivos que el usuario posee en todos sus dispositivos y en sus cuentas en la nube. De esta forma, en lugar de ir conectando memoria USB por memoria USB, disco duro externo uno tras otro y encendiendo un computador tras otro para encontrar el archivo, basta con realizar una búsqueda desde cualquier computador o a través de la aplicación web a la que se puede acceder desde un dispositivo móvil como el teléfono celular.

PJCat está diseñada como aplicación de escritorio desarrollada en Java y aplicación web desarrollada en PHP como lenguaje de lado servidor. Al instalar y ejecutar la aplicación de escritorio en un computador, ésta detecta todos los dispositivos conectados al mismo (memorias USB, discos externos, etc.) y muestra todos los directorios disponibles para que el usuario pueda elegir cuál quiere registrar en la base de datos. Una vez realizado este proceso, se puede buscar cualquier archivo que se encuentre en estos dispositivos sin necesidad de tenerlos conectados. Al acceder a la aplicación web desde cualquier dispositivo, se puede conectar a las cuentas de Dropbox y Drive del usuario, y con ello también se puede realizar búsquedas sobre las mismas.

Una vez sincronizados todos los dispositivos que el usuario desee con la aplicación de escritorio y enlazadas las cuentas de nube de Dropbox y/o Drive con la aplicación web, el usuario puede realizar búsquedas sobre cualquiera de las dos aplicaciones, y los resultados se podrán visualizar tanto de dispositivos físicos como de cuentas en la nube indicando la ruta en la cual el archivo encontrado se encuentra localizado.

Se realizaron pruebas de escritorio para comprobar la rapidez del sistema, ya que se toma en consideración que este problema de buscar archivos consume horas útiles

para las personas. Se encontró que incluso en casos de directorios muy poblados y que arrojen bastantes resultados, la aplicación no toma más de unos segundos en mostrar todo lo que el usuario buscó, indicando el dispositivo o cuenta de la nube y la ruta desde la raíz del mismo. Se considera de esta forma que la propuesta de PJCat es altamente beneficiosa para el usuario final, pues resuelve a cabalidad la problemática presentada y otorga la ventaja de ser multi-dispositivo ya que la aplicación web es accesible desde cualquier computador, celular o tableta con acceso a un navegador.

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA.....	v
RESUMEN	vi
ÍNDICE GENERAL.....	viii
CAPÍTULO 1	1
1. DIFICULTAD PARA ENCONTRAR ARCHIVOS EN MÚLTIPLES DISPOSITIVOS.....	1
CAPÍTULO 2.....	3
2. PJCAT: SISTEMA DE CATÁLOGO PARA REALIZAR BÚSQUEDAS	3
2.1. Diseño de la solución propuesta.....	5
2.2. Implementación de la solución propuesta	7
3. RESULTADOS DE USO DE PJCAT.	12
3.1. Medición de tiempos de búsquedas	13
CONCLUSIONES Y RECOMENDACIONES	16
BIBLIOGRAFÍA.....	17
ANEXO	19

CAPÍTULO 1

1. DIFICULTAD PARA ENCONTRAR ARCHIVOS EN MÚLTIPLES DISPOSITIVOS.

Hoy en día, es muy común que gran parte de la población tenga por lo menos un dispositivo de almacenamiento de archivos, sea este el disco duro de su computador, o un disco extraíble como las memorias USB. Muchas veces, debido a la cantidad de archivos que almacenamos en algún que otro dispositivo, nos cuesta recordar en qué directorio exactamente se encuentra el archivo al que deseamos acceder. Sin embargo, muchos sistemas operativos proveen formas de búsqueda en cualquier directorio, eliminando así este problema. De esta forma, las personas que almacenan todos sus archivos en un mismo dispositivo pueden acceder a cualquier archivo que necesiten en cualquier momento buscándolo.

A pesar de las facilidades que nos dan los sistemas operativos para realizar búsquedas, con el desarrollo y disminución de costo de la tecnología, es cada vez más común que los usuarios de computadores tengan más de un sistema de almacenamiento de archivos. Pueden tener varios discos en su computador, discos externos, dispositivos de almacenamiento extraíble como memorias USB, tarjetas de memoria, e incluso cuentas en la nube con sistema de almacenamiento masivo como Dropbox, Google Drive, entre otras.

Si bien es cierto que cualquiera de los dispositivos físicos antes mencionados puede conectarse a un computador y con la ayuda del sistema operativo se pueden realizar las búsquedas, así como también se puede buscar en cuentas de la nube gracias a la ayuda brindada por el propio sistema online, también es cierto que para llevar a cabo esta tarea habría que ir buscando en cada dispositivo y cuenta en la nube por separado. Y en el caso de los dispositivos físicos habría que irlos conectando al computador para ir buscando uno por uno donde se encuentra el archivo que se está tratando de hallar.

Intentar encontrar un archivo en medio de múltiples dispositivos de almacenamiento, tanto físicos como en la nube, es el equivalente digital de encontrar una aguja en un

pajar. Jason Shechtman, gerente de desarrollo de negocios en la empresa Senexx, indica que el *tiempo promedio* que invierten sus empleados buscando la información que necesitan es de aproximadamente 1.8 horas al día, lo que equivale a 9.3 horas a la semana. Es decir, se considera que si contrata a cinco personas, básicamente se tiene a cuatro personas trabajando y a la quinta buscando respuestas pero sin contribuir con el valor agregado que podría aportar un empleado [1]. Según una encuesta realizada por Osterman Research, los empleados de grandes empresas tienden a demorarse alrededor de 38 minutos sólo en encontrar un documento que se encuentre en sus computadores [2].

En la actualidad, tanto empresarios como particulares buscan optimizar productividad, sea la de los empleados en empresas como la de uno mismo. Siempre se busca seguir la filosofía de “hacer más con menos recursos” y el tiempo es un recurso clave. Fallar en encontrar un documento importante o demorar mucho en ello es una clara muestra de ineficiencia que no puede ser tolerada en el mundo actual y para la cual se necesita una solución que recorte estos tiempos de búsqueda y que en la medida de lo posible no requiera revisar uno por uno los dispositivos físicos sino que de alguna forma su información se encuentre catalogada e indexada.

CAPÍTULO 2

2. PJCAT: SISTEMA DE CATÁLOGO PARA REALIZAR BÚSQUEDAS.

En el capítulo anterior, se mencionó la existencia del problema de tiempo perdido buscando un archivo o conjunto de archivos en múltiples dispositivos de almacenamiento, sean estos físicos como discos duros, memorias USB, tarjetas de memoria, o en la nube como cuentas de Dropbox, Google Drive, entre otras. A continuación mostraremos la solución planteada para dicho problema.

En vista de la diversidad de lugares de donde pueden venir los archivos, hemos considerado la idea de crear un sistema que indexe y catalogue los mismos. De esta forma, el sistema “leería” un directorio y determinaría las rutas de todos los archivos o carpetas en el mismo y los hijos que los componen. Acto seguido, se guardaría en un sistema de base de datos un registro para cada archivo, incluyendo su nombre, su ruta, y un identificador del dispositivo (*nombrado por el usuario para fácil identificación*). Con esto, podríamos buscar el nombre de un archivo y obtener todas las posibles coincidencias, indicando en qué dispositivo se encuentra y en qué ruta del mismo.

A primera vista, el sistema de base de datos cubriría todo, sin embargo, existen alternativas para los sistemas de almacenamiento masivo en la nube, pues tanto Dropbox como Google Drive permiten búsquedas online de sus archivos cuando uno accede a dichos sistemas. Estas búsquedas son optimizadas y utilizarlas eliminaría carga innecesaria de la base de datos, a la vez que mejora los tiempos de respuesta. Para llevar a cabo el uso de las búsquedas optimizadas de Dropbox y Google Drive, se considera la utilización de los API de búsqueda de ambos sistemas. Dichos APIs vienen en diversos lenguajes de programación para acoplarse a las necesidades del proyecto. [3] [4]

Por otro lado, para efectos de facilitar la lectura de todos los directorios de un dispositivo físico, se considera el uso de una aplicación de escritorio, ya que ésta

puede acceder a los recursos del computador y como tal, puede aumentar la eficiencia al momento de barrer todas las carpetas y archivos de algún dispositivo.

Elaborar una aplicación de escritorio implica entonces que una vez realizado el barrido de directorios, la búsqueda se podría hacer únicamente en ese mismo computador o en algún otro que también instale la aplicación. Para solventar este problema y brindarle ayuda al usuario con el uso de múltiples dispositivos, se plantea el desarrollo de una aplicación web, la cual no podrá acceder a todos los recursos del computador y por ende no podrá ser utilizada para el registro de datos de dispositivos físicos, pero permitirá realizar las búsquedas. Se toma en consideración el uso de diseño “responsive” para que la aplicación web se adapte a las diferentes resoluciones de pantalla y con esto pueda ser utilizada desde un teléfono móvil, tableta, computador, o cualquier dispositivo con un navegador web. La figura 2.1 muestra un breve esquema gráfico de cómo se comunicarán los componentes explicados.

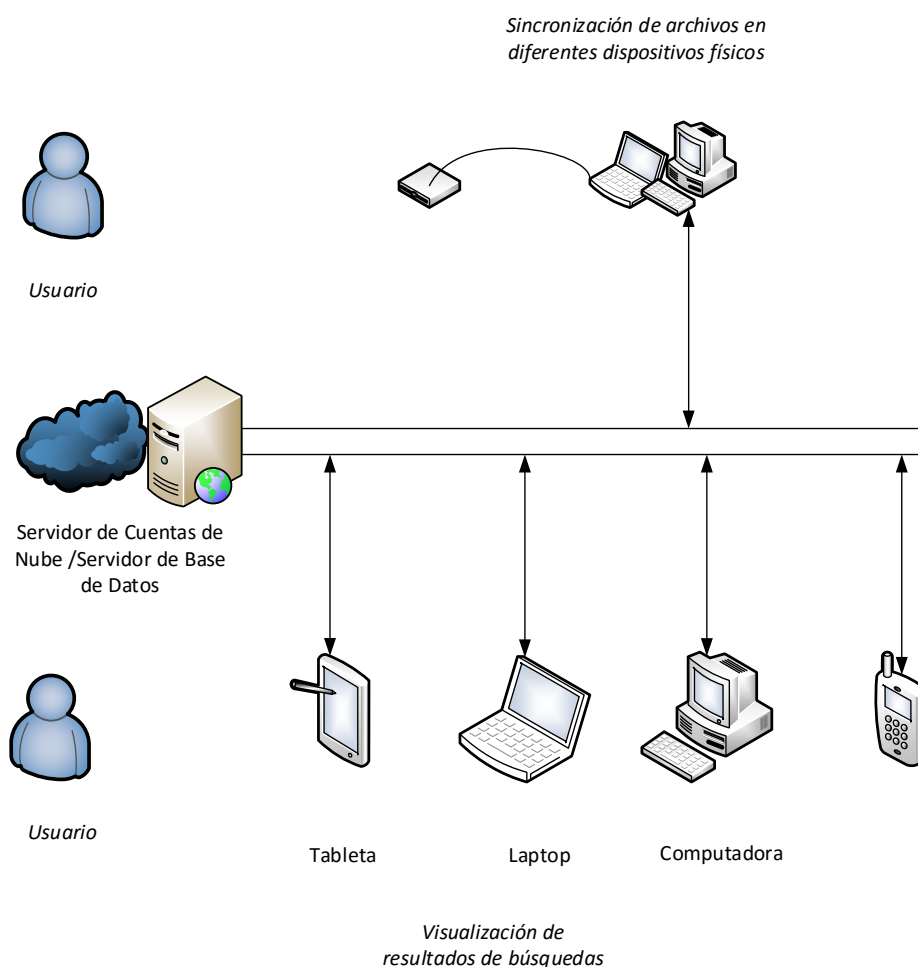


Figura 2.1: Esquema de la solución propuesta

Para poder acceder a las cuentas de almacenamiento en la nube se requiere que el usuario tenga una cuenta para luego ingresar y autorizar la aplicación. El proceso interno de esta autorización consiste en que luego de que el usuario ha realizado la autorización se obtiene un “token de autorización”, el cual es utilizado para las llamadas al API de búsqueda [5] [6] [7] [8].

Para que esto sea posible, es necesario realizar dicho ingreso a través de la web, motivo por el cual el registro de cuentas de almacenamiento en la nube no entra en consideración para la aplicación de escritorio. Es decir, es un requisito fundamental para la aplicación web. Cabe indicar que los servidores de cuenta en la nube seleccionados para este proyecto son: Dropbox y Google Drive.

A través de la solución planteada en este capítulo, se estima que se puede mejorar considerablemente la eficiencia de las búsquedas de archivos por parte de los usuarios, sean estos empleados o personas naturales. Para poder hacer uso del sistema, dichos usuarios sólo necesitarán descargar la aplicación de escritorio y sincronizar archivos de sus dispositivos, y adicionalmente ingresar a la aplicación web para permitir acceso a las cuentas de almacenamiento en la nube. Hecho esto, el usuario podrá utilizar cualquiera de las dos aplicaciones, ingresar con su usuario y contraseña, y realizar búsquedas eficientes de alguna palabra clave, lo que le devolverá en pocos segundos todas las coincidencias, dónde se las encontró y la ruta completa.

2.1. Diseño de la solución propuesta

Como se dijo anteriormente, el proyecto consta de 2 componentes: componente web y componente para computadoras. El componente web se desarrolló usando como lenguaje de programación PHP v. 5.4.3 para lo que es el back-end. Del mismo modo el API de búsqueda para Dropbox (v.1.1.3) y Google Drive (v.1.1.5) utilizado para PHP. Por otro lado la aplicación de escritorio se desarrolló en Java 8 y el gestor del sistema de base de datos utilizado fue MySQL.

La aplicación consta de un diseño sencillo, la cual se encarga de hacer las consultas a la base de datos³ para encontrar coincidencias en los nombre de archivos de los dispositivos sincronizados utilizando la aplicación de escritorio. Además, como se mencionó anteriormente, para acceder a las funciones de búsqueda de archivos en sistemas de almacenamiento en la nube (Dropbox y Google Drive) se requiere un identificador de usuario que haya sido autorizado para acceder a dichas funciones. Este identificador permite a sí mismo recordar que el usuario autorizó el uso de estas funciones, para así no solicitar permisos repetidamente. Es por eso que este identificador o “token” es almacenado en la base de datos, tan pronto se obtiene durante la vinculación del sistema con los sistemas de almacenamiento en la nube antes mencionados.

La aplicación de escritorio, a través de los APIs en java, también realiza las búsquedas sobre los sistemas de almacenamiento en la nube. Esta aplicación es además la encargada de la indexación de archivos en dispositivos físicos. Es por eso que el usuario debe acceder con su cuenta y seleccionar los directorios de los dispositivos conectados a su computador, con el fin iniciar el catalogado de los archivos. Cabe indicar que el usuario es capaz de etiquetar cada dispositivo con un nombre a su elección (Ej: USB HP Azul, Laptop HP Negra).

Luego de que se finalice el catalogado en la base de datos, se mantendrá activo un continuo monitoreo de cambios en los directorios seleccionados, de tal forma que el catálogo de archivos se mantenga siempre actualizado. Si por alguna razón el proceso de catalogado se detiene, éste continuará automáticamente en la siguiente ejecución de la aplicación.

Si se ha realizado el catalogado de un directorio en un dispositivo de almacenamiento extraíble, como una memoria USB, la aplicación detectará automáticamente cuando dicho dispositivo se conecte a la computadora mediante el etiquetado asignado inicialmente por el usuario. Una vez detectada la conexión del dispositivo, el sistema

³ Ver Anexo: “DISEÑO ESQUEMÁTICO DE LA BASE DE DATOS RELACIONAL UTILIZADA”

procede a revisar si existen cambios, ya sea de archivos nuevos o archivos eliminados para proceder a actualizar el catálogo.

Para la identificación de dispositivos ya catalogados como el disco local del computador, se generaron dos archivos que contienen “metadatos”. Un archivo es el que tiene la información sobre el dispositivo, es decir el etiquetado que le dio el usuario, y el otro archivo tiene almacenado el listado de directorios ya catalogados del dispositivo. Para la reanudación del proceso de catalogado, se tomó como ejemplo el esquema observado en la aplicación de escritorio de Dropbox. Es decir primero se descarga un listado de los archivos ya existentes en los directorios catalogados, para proceder a comparar con los archivos existentes en los mismos directorios, de tal manera que se verifica si hay nuevos archivos o si se han eliminado archivos. De esta forma se actualiza el catálogo de archivos alojado en el sistema de base de datos.

Una vez revisado el diseño del sistema se procedió a realizar la implementación del mismo.

2.2. Implementación de la solución propuesta

Para el fácil acceso a las búsquedas y para realizar pruebas de rendimiento, se optó por conseguir hosting para alojar la aplicación web con soporte PHP y al mismo tiempo para alojar la base de datos MySQL. Microsoft Azure ofrece servicios web y diversas plataformas para diferentes usos, uno de ellos es ofrecer servicios para alojar una aplicación web con soporte PHP y una base de datos MySQL [9]. Se escogió usar los servicios de Microsoft Azure debido a que se obtuvo una membresía gratuita de 1 año de Microsoft Azure y porque ofrece alojar una aplicación web con el soporte necesitado⁴.

Se puede acceder a la aplicación web a través de un navegador web y digitar la URL: “<http://pjcat.azurewebsites.net>” cuyo servidor web está ubicado en el sur de Brasil⁵.

⁴ La fecha de inicio de la suscripción de Microsoft Azure por medio de Dreamspark fue el 14/12/2015, es decir es válida hasta el 14/12/2016.

⁵ La ubicación se obtuvo en las propiedades/especificaciones de la suscripción de Microsoft Azure

La figura 2.2 muestra una captura de pantalla del inicio de sesión de la aplicación web alojada en la ubicación mencionada anteriormente (imagen del lado izquierdo) y una captura de pantalla del inicio de sesión desde la aplicación de escritorio (imagen del lado derecho).

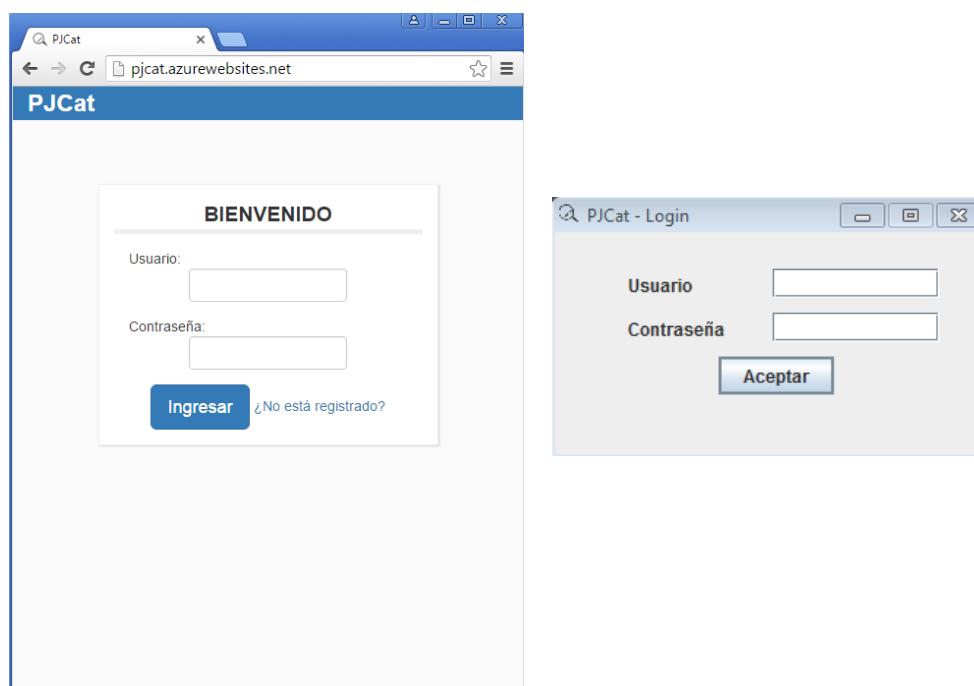


Figura 2.2: Captura de pantalla del inicio de la aplicación web

Cabe indicar que para poder usar tanto la aplicación web como la aplicación de escritorio se debe tener una cuenta registrada. Esto se lo realiza mediante la opción “No está registrado” a través de la aplicación web. Por otro lado la aplicación de escritorio permite el acceso únicamente cuando ya se tenga las credenciales de usuario-contraseña para acceder a las funcionalidades de la aplicación de escritorio mencionadas en el diseño de la misma.

La figura 2.3 muestra que una vez ingresado exitosamente al sistema mediante la aplicación de escritorio, se muestran 2 pestañas. La primera pestaña (imagen del lado izquierdo) muestra la sección de búsquedas que, como se mencionó anteriormente, también se pueden realizar a través de esta aplicación. Por otro lado, la segunda pestaña (imagen del lado derecho) muestra la selección de directorios de los

dispositivos conectados en el actual computador para la pertinente sincronización del catálogo de archivos.

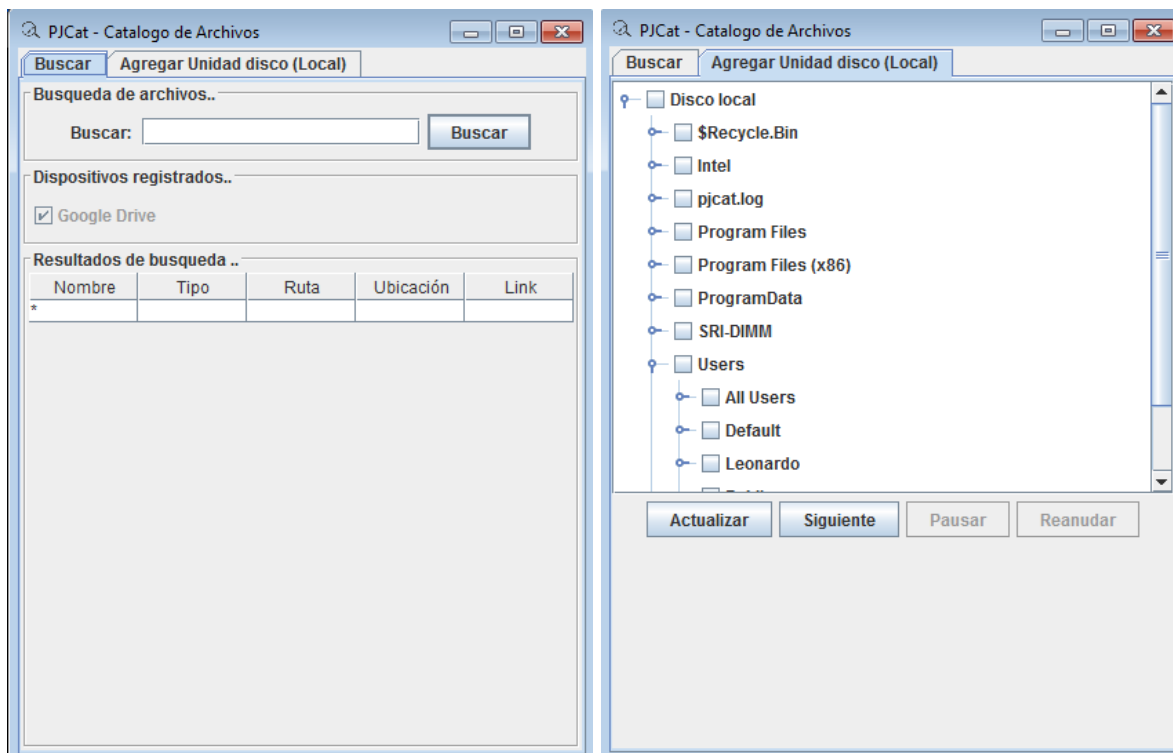


Figura 2.3: Captura de pantalla de la aplicación de escritorio

De esta manera, desde la pestaña “Agregar Unidad disco (Local)” aparecen las unidades de disco conectados como “raíz” del árbol de directorios tal como se puede observar en la imagen del lado derecho de la figura 2.3. Cabe indicar que si se conecta o se desconecta alguna unidad extraíble, se puede actualizar la estructura de árbol utilizando el botón “Actualizar”. Cuando ya se hayan seleccionado los directorios a utilizar para el catalogado, se procede a presionar el botón “Siguiente”. Los botones de “Pausar” y “Reanudar” permiten respectivamente pausar el proceso de catalogado y reanudarlo después de un tiempo deseado. Por otro lado, si se cierra la aplicación o se apaga el computador utilizado, la siguiente vez que se ejecute la aplicación, ésta procederá a sincronizar y catalogar lo faltante para tener todo el catálogo completo

Una de las funcionalidades de la solución propuesta es poder vincularse con cuentas de almacenamiento en la nube, en este caso se seleccionaron Dropbox y Google Drive. Para poder hacer la vinculación, se accede a la aplicación web y desde el menú desplegable se selecciona la opción “Cuenta”. En dicha opción, a más de poder modificar datos personales, se tiene la opción de vincular o desvincular la cuenta de PJCat con una cuenta de Dropbox o de Google Drive. La figura 2.4 muestra una captura de pantalla del menú desplegable⁶ de la aplicación con las diversas opciones que hay en la aplicación web (imagen del lado izquierdo). También se muestra en la misma figura (imagen del lado derecho) el contenido de la sección “Cuenta”, donde se puede vincular y desvincular una cuenta de Dropbox y una cuenta de Google Drive, además de modificar datos de la cuenta de PJCat, como se mencionó anteriormente.

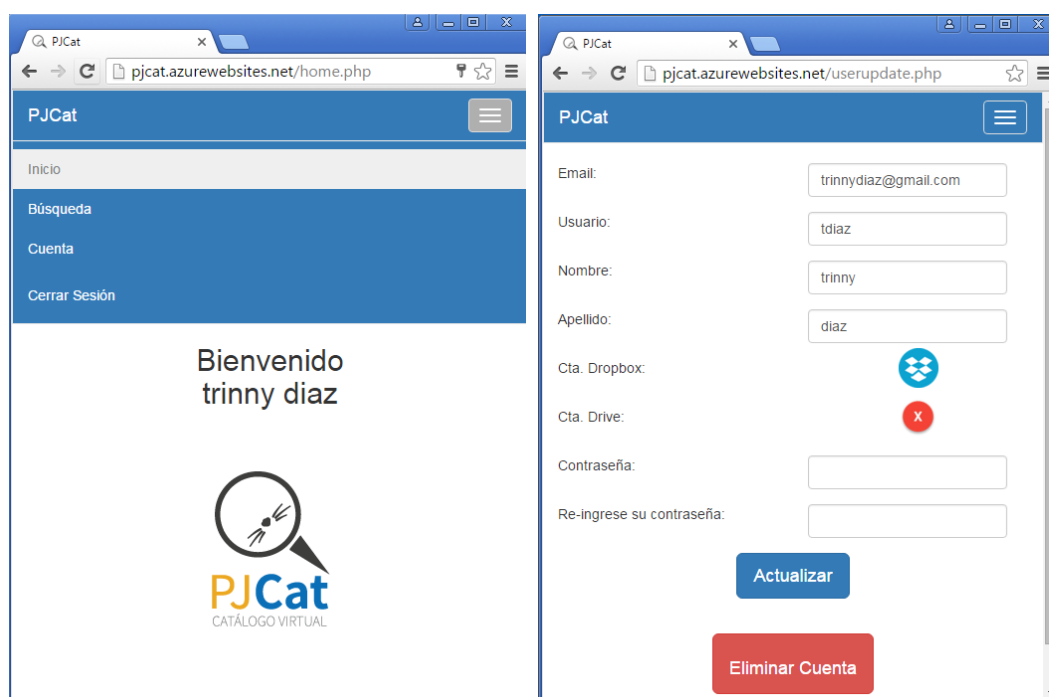


Figura 2.4: Captura de pantalla de la aplicación web de la sección “Cuenta”

Finalmente, la funcionalidad de las búsquedas se la puede apreciar en la figura 2.5. La aplicación web permite visualizar los dispositivos sincronizados y a su vez permite darle de baja a cualquiera de ellos. Los resultados en la misma, se desglosan en 3

⁶ El menú desplegable aparecerá para pantallas con resolución <768px. De lo contrario aparece cada opción del menú en una sola línea

diferentes tablas (imagen del lado izquierdo). Una tabla muestra los resultados de los dispositivos sincronizados, una segunda muestra los de Dropbox y la última los de Google Drive. En las tablas de los resultados de Dropbox y Google Drive aparecen enlaces para descargar dichos resultados si el usuario así lo desea. Por otro lado en la misma figura (imagen del lado derecho) se muestra una captura de pantalla de las búsquedas a través de la aplicación de escritorio, en cuyo caso los resultados se muestran en una sola tabla. Se optó por desglosar los detalles de los resultados en la aplicación web porque la solución propuesta está destinada a ser multi-dispositivo y por ello eso se realiza con mayor comodidad para el usuario a través la web

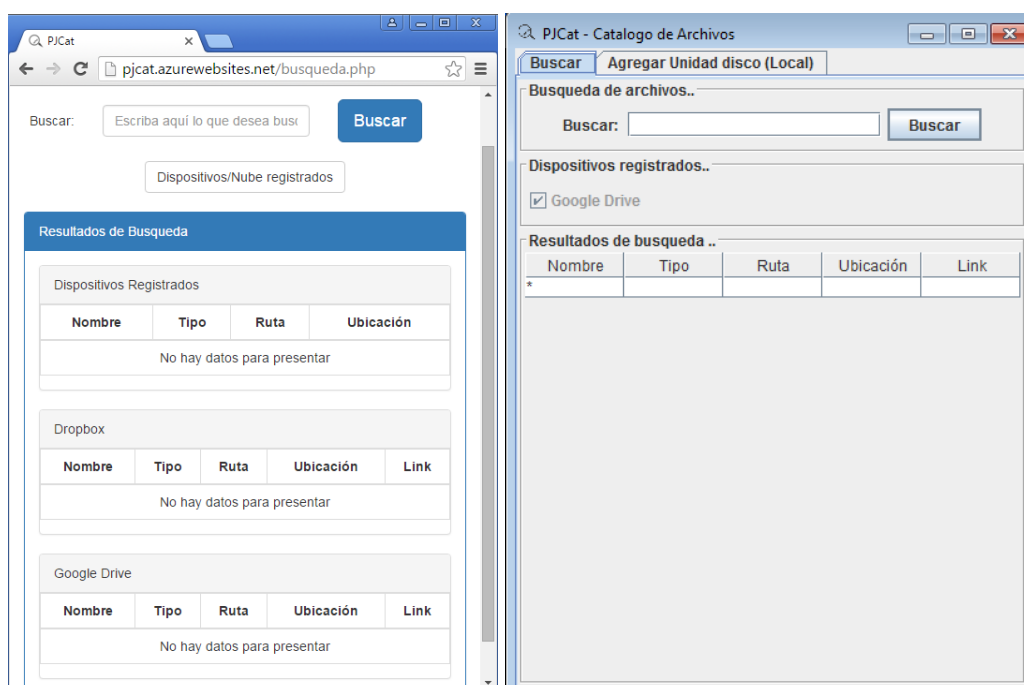


Figura 2.5: Captura de pantalla de la aplicación web de la sección “Búsqueda” y de la aplicación de escritorio de la pestaña “Buscar”

CAPÍTULO 3

3. RESULTADOS DE USO DE PJCAT.

Como se mencionó en el capítulo anterior, PJCat consta de dos partes: una aplicación de escritorio y una aplicación web. Ambas aplicaciones permiten realizar búsquedas en los sistemas de almacenamiento del usuario ingresado, sean sistemas físicos o en la nube. Por su parte la aplicación de escritorio permite agregar nuevos dispositivos físicos para que se pueda buscar en ellos mientras que la aplicación web permite enlazar o desenlazar una cuenta de Dropbox o de Google Drive.

Para que el sistema pueda ser visto desde cualquier dispositivo y desde cualquier lugar, tanto los archivos de la aplicación web como el sistema de base de datos se alojan en un servidor al que se puede acceder por Internet. Esto implica que para obtener mejores resultados, una rápida conexión a Internet es imprescindible.

Se realizaron pruebas de rendimiento del sistema considerando cuentas de Dropbox, Google Drive y un disco duro de computador. Se considera un típico estudiante de Ciencias Computacionales que guarda en algunos directorios de su computador y en la nube sus proyectos, por lo cual tiene carpetas con muchos subniveles. En todos los casos, por más que se busquen nombres repetidos, se lograba obtener el detalle de todos los resultados en menos de un minuto.

A diferencia del sistema implementado por nosotros, en que cada archivo o carpeta tendría asignada su ruta, y a diferencia también del API de Dropbox, que también devuelve la ruta, el API de Google Drive no devuelve la misma, pero devuelve un identificador de la carpeta padre, con lo que se pudo elaborar un algoritmo recursivo que arroje la ruta. Realizando pruebas respectivas, se pudo obtener la ruta de todos los elementos que se buscaban en Google Drive, pero cuando se buscó una palabra que arrojaba más de 60 resultados, los cuales están todos en una misma ruta, el sistema demora sustancialmente. Para evitar dicha demora implementamos un diccionario de rutas con lo que sólo se ejecutaría este algoritmo recursivo una vez por directorio. Al implementar esto, nuevamente el sistema muestra respuestas dentro de los primeros 20 a 30 segundos para carpetas bastante pobladas en Google Drive sin considerar paginar dichos resultados.

3.1. Medición de tiempos de búsquedas

La medición de tiempos de búsqueda realizada fue hecha a nivel de la aplicación web, considerando que el usuario accederá más veces a la aplicación web que a la aplicación de escritorio para realizar búsquedas. Para dicha medición, se utilizó como navegador Google Chrome (v. 49.0.2623.75 m), deshabilitando todas las extensiones que tenga el navegador instaladas⁷. La herramienta de desarrollador que ofrece el navegador de Google Chrome permite visualizar la actividad de la red de una página web mediante la pestaña “red” de dicha herramienta [10]. Se puede ver en la figura 3.1 una captura de la herramienta ya mencionada con una actividad que se refleja al momento de hacer 3 búsquedas diferentes (buscando resultados para: “aguacate”, “guayaba” y “archivo” respectivamente). Dicha herramienta permite visualizar el tiempo que se toma en recibir una respuesta para cada llamada de búsqueda, en cuyo caso las recibe el archivo “getResultadosBusqueda.php”

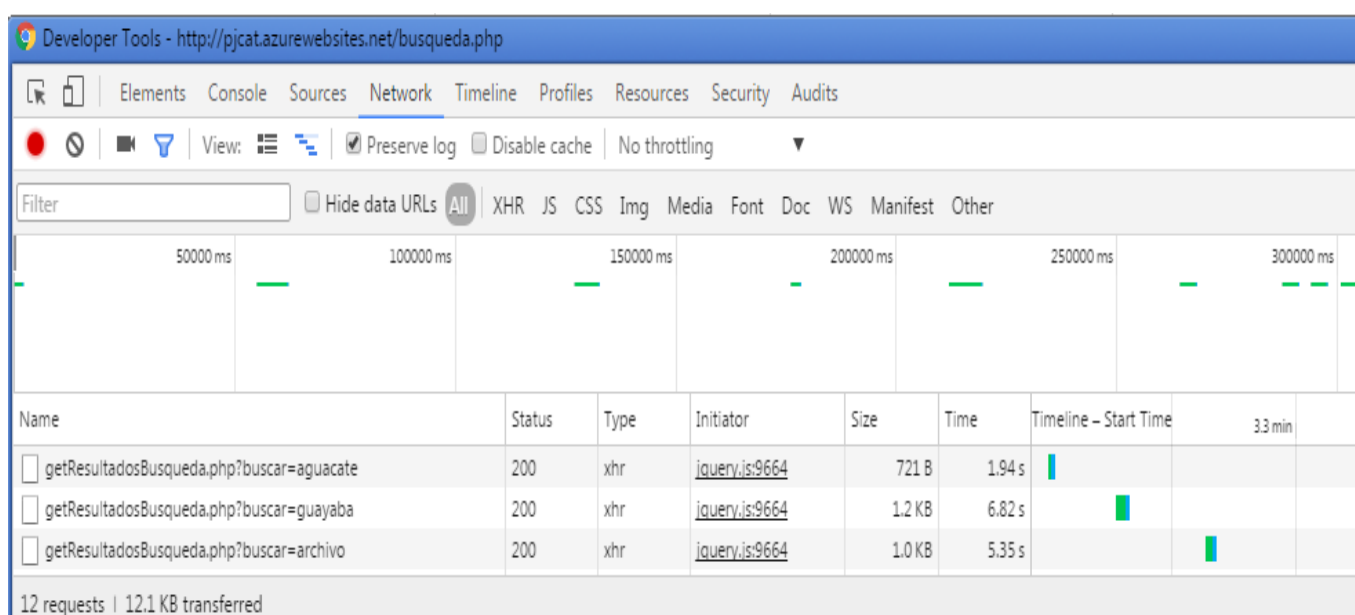


Figura 3.1: Captura de pantalla de la herramienta de desarrollador de Google Chrome para visualizar tiempos de respuesta para diferentes casos de búsqueda

⁷ Se deshabilitó las extensiones con el fin de evitar que afecten los tiempos de búsqueda a nivel del navegador web.

De esta manera se procedió a realizar pruebas de rendimiento para hacer comparaciones entre tiempos de búsqueda de la aplicación web en comparación la encuesta realizada por Osterman Research y lo mencionado por el gerente de Senexx. Dichos resultados se presentan en la tabla 1 a continuación.

Valor de búsqueda	¿Hubo Resultados?	Resultados					Tiempo de Búsqueda
		Dispositivos Sincronizados		Dropbox	Google Drive		
		Paginas	Archivos	Archivos	Páginas	Archivos	
<i>aguacate</i>	No	-	-	-	-	-	1.94s
<i>guayaba</i>	Si	-	-	-	1	5	6.82s
<i>archivo</i>	Si	-	-	3	-	-	5.35s
<i>anexo</i>	Si	1	1	-	-	-	2.01s
<i>imagen</i>	Si	-	-	2	8	39	7.32s
<i>java</i>	Si	138	687	244	-	-	3.5min
<i>inteligencia</i>	Si	1	1	3	-	-	4.06s
<i>introducción</i>	Si	1	3	8	-	-	8.72s
<i>proyecto</i>	Si	1	1	42	-	-	29.17s
<i>.jar</i>	Si	20	100	22	-	-	22.68s
<i>tutorial</i>	Si	1	1	5	-	-	7.08s
<i>.iso</i>	Si	-	-	-	1	1	5.65s
<i>.pdf</i>	Si	2	6	141	1	5	2.2min

Tabla 1: Tabla de resultados de tiempos de búsqueda para diferentes casos usando la aplicación web

Se puede apreciar que la mayoría de resultados tiene tiempos de búsqueda por debajo de 10 segundos, sin embargo existen tiempos por encima de 1 minuto. Cabe hacer hincapié que estos casos cuyos tiempos de búsqueda superan el minuto de duración, contienen una gran cantidad de archivos encontrados en Dropbox, sistema cuya API no permite paginación de resultados por ahora, a diferencia de Drive que sí lo permite⁸. Para las búsquedas en Google Drive y de los dispositivos registrados se realiza una paginación de los resultados, en este caso se eligió 5 resultados por páginas. Cabe indicar que los tiempos de búsqueda se ven afectados por la respuesta

⁸ La versión 2 del API de Dropbox si permite paginación, sin embargo para PHP no está disponible y por ello se utilizó la versión 1.

del servidor, lo cual incluye la ubicación física del mismo, que está ubicado en Brasil y a su vez la velocidad del proveedor de internet [11] [12].

Realizando un análisis sobre la problemática, el sistema permite hacer búsquedas en menos de un minuto (en promedio) y mostrando la ruta de cada coincidencia para que el usuario final pueda corroborar. En otras palabras, estas búsquedas realizadas a través de la solución propuesta ahorran muchísimo tiempo y permiten aprovechar de manera apropiada la productividad del usuario final.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. A través del uso de PJCat se logró eficientemente mejorar el rendimiento de trabajadores y disminuir los tiempos de búsqueda de archivos en múltiples dispositivos.
2. Se ha logrado una exitosa indexación, integración de archivos en tanto dispositivos físicos como en sistemas de almacenamiento en la nube, permitiendo búsquedas eficientes desde cualquier dispositivo en cualquier parte del mundo a través de la aplicación web.
3. Se ha logrado reconocer dispositivos ya sincronizados y reanudado el catalogado, permitiendo una mejor experiencia de usuario y a su vez para tener siempre actualizado el catálogo de archivos

Recomendaciones

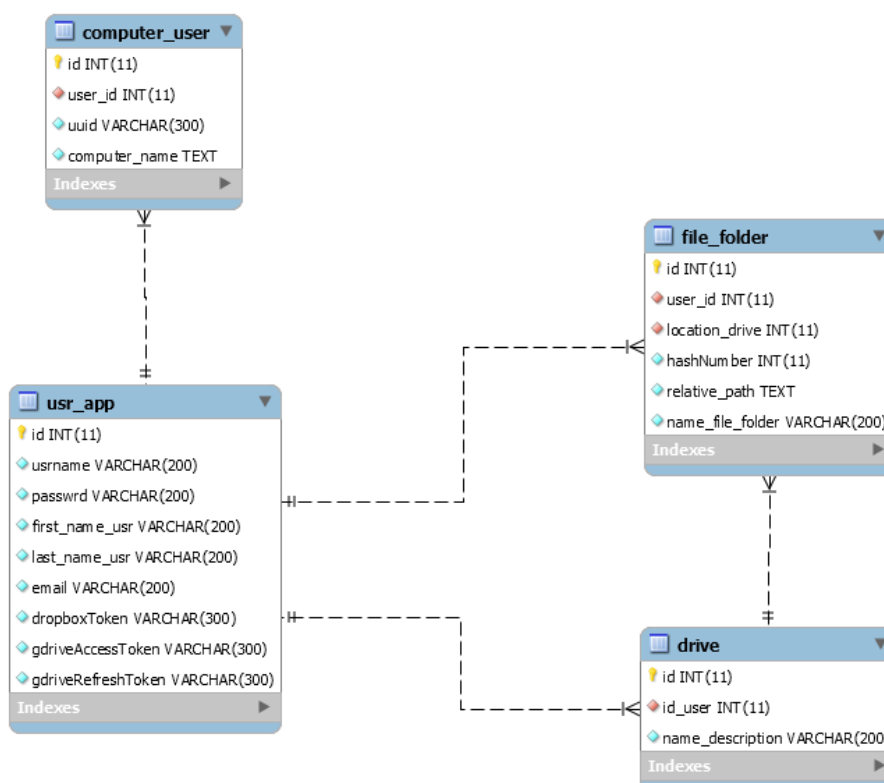
1. Se recomienda como trabajo a futuro poder utilizar APIs para Java y PHP que permitan la integración con otros sistemas de almacenamiento en la nube, permitiendo así al usuario utilizar sistemas como OneDrive, MEGA, entre otros.
2. Se considera para futuros lanzamientos del sistema, almacenar los archivos de los dispositivos físicos del usuario, para así permitirle descargarlos desde la web.
3. Se toma en consideración el uso de bases de datos no relacionales para mejorar tiempos de búsquedas en caso de que el sistema crezca a gran escala.

BIBLIOGRAFÍA

- [1] Burgee, "Top 3 Reasons Why We Spend So Much Time Searching for Information" Mayo 2013. [En línea]. Disponible en: <http://utrconf.com/top-3-reasons-why-we-spend-so-much-time-searching-for-information/>. [Último acceso: 23 Febrero 2016].
- [2] Wailgum, "Have You Ever Spent 38 Minutes Looking for a Document?" Mayo 2009. [En línea]. Disponible en: <http://www.cio.com/article/2428183/enterprise-software/have-you-ever-spent-38-minutes-looking-for-a-document-.html>. [Último acceso: 23 Febrero 2016].
- [3] Google, "Google Drive REST API Overview" [En línea]. Disponible en: https://developers.google.com/drive/v3/web/about-sdk#create_and_open_files_directly_from_the_drive_ui. [Último acceso: 26 Octubre 2015].
- [4] Dropbox, "Dropbox Platform developer guide" [En línea]. Disponible en: <https://www.dropbox.com/developers/reference>. [Último acceso: 26 Octubre 2015].
- [5] Dropbox, "Using the Core API in PHP" [En línea]. Disponible en: <https://www.dropbox.com/developers-v1/core/start/php>. [Último acceso: 26 Octubre 2015].
- [6] Google, "PHP Quickstart - Drive REST API" [En línea]. Disponible en: <https://developers.google.com/drive/v3/web/quickstart/php>. [Último acceso: 26 Octubre 2015].
- [7] Dropbox, "Using the Core API in Java" [En línea]. Disponible en: <https://www.dropbox.com/developers-v1/core/start/java>. [Último acceso: 26 Octubre 2015].
- [8] Google, "Drive API Client Library for Java" [En línea]. Disponible en: <https://developers.google.com/api-client-library/java/apis/drive/v2>. [Último acceso: 26 Octubre 2016].
- [9] Microsoft, "Web App Service" Microsoft, [En línea]. Disponible en: <https://azure.microsoft.com/en-us/services/app-service/web/>. [Último acceso: 26 Octubre 2015].

- [10] Google, "Measure Resource Loading Times" [En línea]. Disponible en: <https://developers.google.com/web/tools/chrome-devtools/profile/network-performance/resource-loading>. [Último acceso: 26 Octubre 2015].
- [11] A. P.-D. Medi, "How Does Bandwidth Affect Website Performance?" [En línea]. Disponible en: <http://smallbusiness.chron.com/bandwidth-affect-website-performance-69958.html>. [Último acceso: 26 Octubre 2015].
- [12] M. Mittal, "15 Website Performance Indicators You Should Monitor" [En línea]. Disponible en: https://www.es.paessler.com/blog/2011/05/13/monitoring-knowledge/15_website_performance_indicators_you_should_monitor. [Último acceso: 26 Octubre 2015].

ANEXO

DISEÑO ESQUEMÁTICO DE LA BASE DE DATOS RELACIONAL UTILIZADA.

La imagen refleja el esquema de base de datos relacional que es bastante sencillo para el catalogado a realizar.