

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

AUTOMATIZACIÓN Y PARAMETRIZACIÓN DE PROCESOS DE BÚSQUEDA Y RECUPERACIÓN DE LLAMADAS

DANIEL OSWALDO VEGA CARVAJALINO

Ingeniería Mecatrónica

Juliana Valencia Aguirre

INSTITUTO TECNOLÓGICO METROPOLITANO

04 de agosto 2017

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

En este proyecto se buscó aumentar la eficiencia del área de tecnología y grabación de la empresa Konecta la cual es uno de los Call center mas importantes de america, mediante el desarrollo de dos aplicaciones de software las cuales se encargarán de parametrizar y organizar la información de reportes e inventarios de las llamadas realizadas. Para optimizar la búsqueda y recuperación de las grabaciones de las mismas, ya que esta labor es de alta prioridad para la empresa.

Hasta el momento esta labor se realizaba de forma manual. El proyecto Automatiza el proceso buscando en los distintos servidores la información necesaria para que esta pueda interpretada por una máquina, automatizando a su vez el proceso de búsqueda y recuperación de las llamadas ya que al estar parametrizados estos inventarios es posible crear una base de datos sólida, donde una maquina es capaz de realizar la labor que antes debía ser realizada por un operario reduciendo considerablemente la perdida de información y mejorando el tiempo de respuesta del área.

Palabras clave: Automatización, grabación, red, servidores, base de datos.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

El presente proyecto pudo ser realizado gracias al apoyo de los jefes del área de tecnología y grabación del grupo konecta los cuales dieron el aval para el desarrollo y brindaron todas las licencias para los softwares de desarrollo y bases de datos necesarios para llevar a cabo el mismo. También es necesario agradecer a todo el cuerpo de desarrolladores de la empresa que siempre estuvo dispuesto a brindar las asesorías necesarias a los problemas presentados a la hora de programar cada módulo del proyecto.

También es necesario Agradecer al ITM que dio las bases necesarias para poder llevar acabo el desarrollo del proyecto, en especial a la docente del área de Electromecánica y Mecatrónica Juliana valencia la cual como asesora del proyecto siempre estuvo dispuesta a resolver dudas y ayudar en cada uno de los aspectos del mismo.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

Para tener un mejor entendimiento del proyecto es necesario tener en cuenta los siguientes acrónimos.

IP: Internet Protocol

Vba: Visual Basic for Applications

C#: C Sharp

SQL: Structured Query Language

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

1. INTRODUCCIÓN	6
2. MARCO TEÓRICO	9
3. METODOLOGÍA.....	14
4. RESULTADOS Y DISCUSIÓN.....	28
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	29
REFERENCIAS	30
APÉNDICE.....	31

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

La empresa Konecta es uno de los call center más importantes de Latinoamérica y es el encargado de regular, grabar y almacenar llamadas de grandes empresas como lo son Bancolombia, Postobón, Viva Colombia, entre otros. Dada la importancia de dichas llamadas, estas deben ser almacenadas en su totalidad por un tiempo mínimo de 5 años y es responsabilidad del área de tecnología y grabación garantizar el almacenamiento, calidad y rápida respuesta a las peticiones de las llamadas.

Debido a la cantidad de información que debe regularse, la cual es de 3 a 5 Terabytes diarios, es imposible económica y operativamente el mantener el historial completo accesible en la nube, por lo cual la información disponible en la red no supera el año de antigüedad, el resto de la información es almacenada y cifrada en servidores donde solo el área tiene acceso.

Actualmente el proceso de recuperación y búsqueda de las llamadas almacenadas en los servidores se realiza de forma manual, acudiendo a inventarios y reportes que llevan el control y organización de la información. No obstante, por fallos humanos y poca regulación en estos archivos la recuperación de las llamadas puede tomar aproximadamente 1 hora y en muchos casos no es posible su recuperación. Las aplicaciones de software planteadas y desarrolladas en este proyecto buscan unificar todos los inventarios y reportes para tener un acceso más ágil a la información, de manera que esta pueda ser interpretada fácilmente, a la vez que se logra minimizar el error humano por medio de la automatización y hacer más eficiente el proceso más importante y delicado del área.

El desarrollo planteado se realiza en Vba para estandarizar los archivos deseados. Los archivos se almacenan en una base de datos diseñada en SQL y el proceso de búsqueda se automatiza en un software diseñado en el lenguaje C# el cual se encarga de manipular las bases de datos. Al finalizar el proceso se logró reducir el tiempo de búsqueda de una hora

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

a aproximadamente 2 minutos lo que es una mejora considerable en la efectividad del proceso.

1.1 FORMULACIÓN DEL PROBLEMA

El área de tecnología y grabación tiene como labor principal almacenar todas las llamadas recibidas y realizadas por las empresas asociadas al grupo Konecta garantizando alta calidad y una rápida recuperación de las mismas. Debido a la cantidad de información que se debe almacenar, la labor de recuperar las llamadas es un trabajo demandante que requiere un gran esfuerzo y en ocasiones imposible dado que el proceso se realiza casi en su totalidad de forma manual. Por esta razón la pérdida de información y la lenta respuesta frente a importantes requerimientos es un constante problema, que afecta tanto la imagen del área como la de la empresa Konecta, la cual busca una solución fácil de implementar y que ayude en esta importante labor.

1.2 JUSTIFICACIÓN

El proyecto planteado se realiza a partir de la necesidad de regular, organizar y automatizar los procesos de recuperación de llamadas almacenadas por el grupo Konecta, dado que dicha información se encuentra cifrada y almacenada en servidores para los cuales solo se accede de forma manual haciendo que el proceso de recuperación sea lento, poco eficiente y en ocasiones ineficaz.

La mala regulación y organización de dicha información genera un costo económico a la empresa dado que por cada día de retraso en la recuperación o por la pérdida de una llamada se debe pagar una multa o indemnización a la empresa afectada, la cual puede ser de una suma de hasta 100 millones de pesos.

Dado este problema, se plantea el desarrollo de un conjunto de aplicaciones de software que ayuden a reducir el déficit presente en el área de tecnología y grabación, y reduzca los

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

fallos, para así poder dar una respuesta rápida, disminuir la pérdida de información y minimizar los altos costos generados por dicha actividad.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Automatizar el proceso de búsqueda y recuperación de llamadas en las bases de datos y servidores para mejorar la efectividad en el área de tecnología y grabación de la empresa Konecta Colombia.

1.3.2 OBJETIVOS ESPECÍFICOS.

- Desarrollar un software que permita parametrizar los inventarios realizados a mano para que concuerden con el formato de los inventarios actuales desarrollados por medio de queries en bases de datos específicas
- Desarrollar un software que unifique todos los inventarios existentes en una sola base de datos y cruce la información de estos inventarios con otras tablas para que así sea posible recuperar las llamadas de forma automática.
- Validar el funcionamiento e impacto de los programas desarrollados de manera que sea posible comprobar la mejora en el proceso.
- Estandarizar los archivos con el modelo actual y poder automatizarlos por medio de un software que fue desarrollado para esta labor.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

Para el entendimiento de todas las etapas del proyecto es necesario tener claro los conceptos más utilizados en el mismo:

Storage(almacenamiento): la definición general de Storage suele ser la acción de guardar documentos o información en formatos ópticos o electromagnéticos en un ordenador, por ejemplo en el área de tecnología y grabación los storage se almacenan en la memoria interna de los servidores si la información es reciente o en cintas magnéticas de almacenamiento cuando la información es antigua, estos por lo general suelen ser de una capacidad de almacenamiento de 250 a 500 Tb.

Servidor: es un equipo informático que forma parte de una red y provee servicios a otros equipos cliente. Para los fines del proyecto los servidores son los equipos que almacenan toda la información de valor dentro de la empresa, usualmente estos son almacenados en cuartos especializados que brinden la seguridad necesaria para que estos no sean vulnerables tal como se muestra en la figura 1. (<http://searchdatacenter.techtarget.com>,2016)

Figura 1. Ejemplo de la disposición de los servidores.

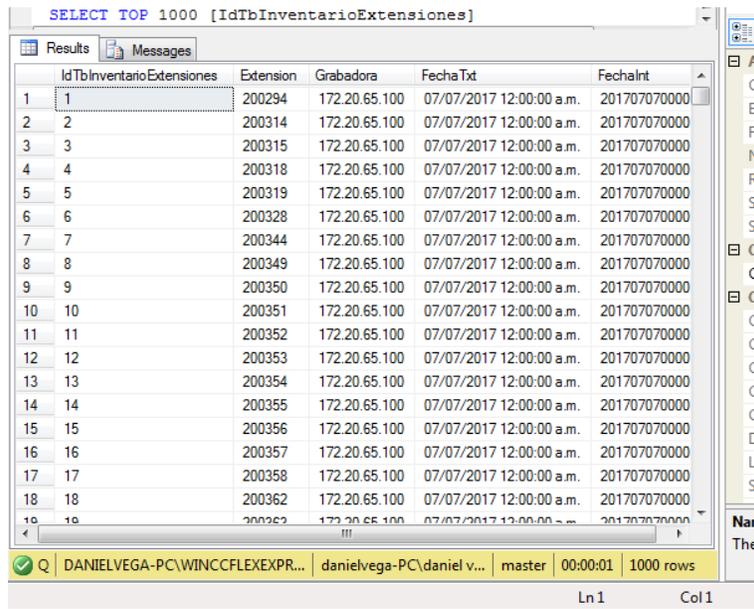


(OkHostinf, (2016),<https://okhosting.com/blog/que-es-servidor>)

Base de datos: desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de información almacenada que permiten el acceso directo a esta y a un conjunto de programas que manipulen ese conjunto de datos.

Una base de datos se compone de una o más tablas que guarda un conjunto de información. cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, las filas de la tabla conforman un registro, como se puede apreciar en la Figura 2 (SQL y desarrollo de aplicaciones en Oracle 8,2001, pág. 2)

Figura 2. Representación tabla SQL.



IdTbInventarioExtensiones	Extension	Grabadora	FechaTxt	FechaInt
1	200294	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
2	200314	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
3	200315	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
4	200318	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
5	200319	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
6	200328	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
7	200344	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
8	200349	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
9	200350	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
10	200351	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
11	200352	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
12	200353	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
13	200354	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
14	200355	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
15	200356	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
16	200357	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
17	200358	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000
18	200362	172.20.65.100	07/07/2017 12:00:00 a.m.	201707070000

Columnas: idTbinventarioExtensiones, Extension, Grabadora , FechaTxt y FechaInt.

Filas: se enumeran consecutivamente y representa todas las características de un individuo de la base (registro)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Automatización: es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.

Un sistema automatizado consta de dos partes principales:

- La Parte Operativa, que es la parte que actúa directamente sobre el proceso. Son los elementos que hacen se realice la operación deseada.
- La Parte de Mando, es lo que se refiere a la parte que decide cuando activar la parte operativa y hace al proceso casi independiente de un operario.

Objetivos de la automatización:

- Mejorar la productividad de un proceso, reduciendo los costes de la producción y mejorando la calidad de la misma.
- Mejorar las condiciones de trabajo del personal, suprimiendo los trabajos penosos e incrementando la seguridad.
- Realizar las operaciones imposibles de controlar intelectual o manualmente.
- Mejorar la disponibilidad de los productos, pudiendo proveer las cantidades necesarias en el momento preciso.

(<http://www.sc.ehu.es/sbweb/webcentro/automatica,2001>)

IP: Una dirección IP es un número que identifica, de manera lógica y jerárquica, a un dispositivo en red que utilice el protocolo TCP/IP. (TCP/IP Foundations, 2004, pág. 2-5)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

VBA: Microsoft VBA es el lenguaje de macros de Microsoft Visual Basic que se utiliza para programar aplicaciones Windows y que se incluye en varias aplicaciones Microsoft, como lo es office Excel el cual contiene funciones y librerías propias en este lenguaje que permiten la fácil manipulación de estos archivos. (<https://docs.microsoft.com>,2017)

C#: es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Su sintaxis básica deriva de C/C++.

Algunas ventajas que ofrece del lenguaje frente a otros lenguajes:

- en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.
- Propiedades: un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.
- Control de versiones: C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

(Iniciación a la programación en C#: un enfoque práctico,2006, pág. 2)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

SQL: es un lenguaje de consulta estructurada que da acceso a un sistema de gestión de bases de datos que permite especificar diversos tipos de operaciones en ellos.

Principales características de SQL :

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

(SQL y desarrollo de aplicaciones en Oracle 8,2001, pág. 2)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

Para el desarrollo del proyecto fueron necesarios amplios conocimientos tanto en programación como en el manejo de bases de datos. Para comenzar fueron necesarias las licencias para realizar los desarrollos y los permisos del área de seguridad de la empresa para instalarlos, además de los permisos y aprobaciones internas de la empresa para utilizar y modificar las bases de datos y la información en los servidores. Para esto se recibió apoyo suficiente por parte de los desarrolladores internos de la empresa y de los encargados de las bases de datos, quienes brindaron apoyo y asesoría durante el desarrollo del proyecto, ya que este no solo debía satisfacer las necesidades y problemáticas del área, sino, que también debía ajustarse a los requerimientos y estándares de la empresa Konecta para que este estuviese aprobado por seguridad y se pudiese manipular la información de dichos servidores.

3.1 DESCRIPCIÓN DEL PROCESO

El proyecto planteado se enfocó en solucionar problemas del área de tecnología y grabación de la empresa Konecta por medio de la automatización de los siguientes procesos:

Se recolectan todos los inventarios y reportes que contienen la información necesaria para la recuperación de llamadas.

Dado que estos archivos se desarrollaban a mano se busca un estándar para reescribir los mismos en un formato más organizado y fácil de interpretar.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se desarrolla un software en Visual Basic 6.0 utilizando el compilador de Excel para organizar los archivos de forma automática estandarizando toda la información almacenada, siendo este capaz de leer e identificar toda la información en estos documentos.

Una vez todos los documentos fueron estandarizados se creó una base de datos en MS-SQL la cual una vez se le importe la información de estos archivos almacenará y realizará las consultas de esta información siendo está más eficiente, organizada y confiable.

Se creó una aplicación de software en C#.net utilizando el compilador Visual Studio el cual ofrece un entorno de consulta de base de datos, automatiza el cruce de las mismas y por último será el encargado de hacer el proceso de interfaz de usuario para la búsqueda y recuperación de las llamadas.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.2 ALCANCES.

Se espera que el proyecto pueda automatizar el proceso de búsqueda y recuperación de llamadas en el área de tecnología de la empresa Konecta.

En primera instancia se pidió una propuesta formal del proyecto esperando que este no pasara de una fase de pre-desarrollo, pero una vez se conoció la ayuda que este proyecto presentaría, se aprobó un desarrollo el cual llegaría a salir a producción siendo así una herramienta necesaria e indispensable en el área; para garantizar los buenos resultados en la búsqueda y recuperación de las llamadas.

3.3 DELIMITACIÓN

3.3.1 DELIMITACIÓN ESPACIAL.

El proyecto desarrollado va enfocado a solucionar una falencia presente en la empresa Konecta Colombia, la cual cuenta con diferentes sedes tanto en Medellín como Bogotá. Los softwares desarrollados serán instalados y puestos en producción en la ciudad de Medellín, lugar en donde se encuentra centralizada toda el área de tecnología y grabación, no obstante, dicho proyecto servirá de apoyo a todas las sedes y áreas de la empresa. Esto con el objetivo de poder ayudar a alcanzar las metas estipuladas en la visión de Konecta la cual habla de “ser la compañía líder global de BPO y Contact Center ofreciendo las mejores experiencias a los clientes” y el acceso eficiente a los historiales es un punto clave para la satisfacción de los clientes.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.3.2 DELIMITACIÓN TEMPORAL.

El tiempo de desarrollo del proyecto fue de 4 meses. Las aplicaciones desarrolladas salieron a producción el 30 de Julio del presente año.

El proceso requiere las siguientes etapas principales:

- Recolección de información.
- Superación de limitaciones.
- Gestión de licencias y permisos.
- Desarrollo.
- Pruebas.
- Salida a producción.

3.4 ASPECTOS ADMINISTRATIVOS

3.4.1 RECURSOS HUMANOS

El desarrollo del proyecto se asignó al practicante del área de tecnología y grabación Daniel Oswaldo Vega Carvajalino, bajo la supervisión del Ingeniero Camilo Andrés Arbeláez y con el apoyo de todos los programadores en el área de desarrollo, los cuales brindaron la asesoría y apoyo necesario para desarrollo del mismo.

3.4.2 RECURSOS MATERIALES

Para el desarrollo del proyecto se utilizó una máquina con suficientes recursos para almacenar la base de datos con toda la información requerida, es necesario tener acceso a la red local de la empresa para que el software pueda ser utilizado por cualquier persona

	INFORME FINAL DE TRABAJO DE GRADO		Código	FDE 089
			Versión	03
			Fecha	2015-01-22

del área y por último tener acceso a los servidores de Konecta para realizar la búsqueda de las llamadas de forma automática.

3.4.3 RECURSOS ECONÓMICOS

Dado que el proyecto es a nivel de software los recursos necesarios son mayormente licencias en los programas de diseño y de bases de datos, se excluye como gasto el de recursos físicos dado que elementos como los computadores o los servidores son elementos con los que la empresa contaba anteriormente.

3.4.4 CRONOGRAMA DE TRABAJO

Para el desarrollo del proyecto se dio un tiempo de 4 meses los cuales se emplearon de la siguiente manera:

Figura 3. Cronograma de trabajo.

Meses	1				2				3				4			
Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Propuesta inicial y estudio de viabilidad	█	█														
Recolección de información		█	█	█	█	█	█									
Gestión de licencias y permisos					█	█	█	█	█	█	█					
Desarrollo									█	█	█	█	█	█	█	█
Pruebas													█	█	█	█
Salida a producción																█

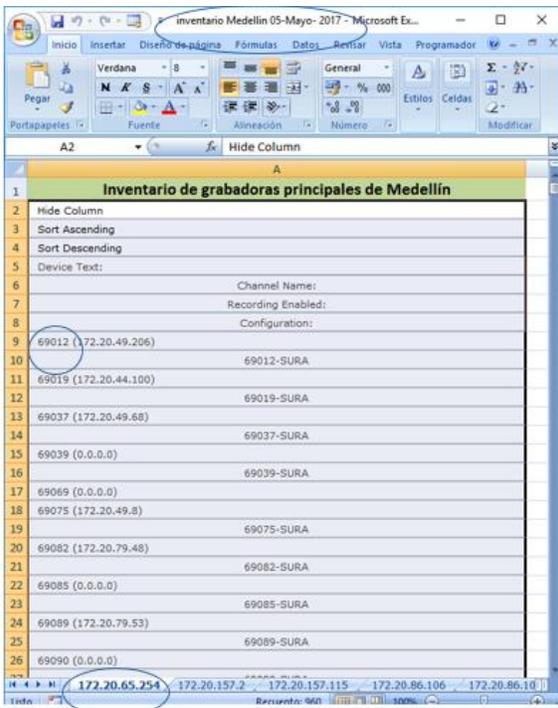
 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.5 SOFTWARE DE PARAMETRIZACIÓN.

Este software busca la estandarización de los inventarios y reportes que hasta el día de hoy se realizaban de forma manual, esto con el fin de facilitar la interpretación y que su formato sea adecuado para ser importados a las bases de datos.

En primera instancia los archivos contenían la información (número de la extensión) en una sola columna del Excel y se encontraba mezclada con información innecesaria como lo es el cliente que trabaja con la extensión, Ip asociada, entre otros. Dichas columnas se encontraban distribuidas en hojas del documento cuyo nombre era del servidor de donde provenía la información como se muestra en la figura4. Donde se puede apreciar como se encontraban la información, volviendo un proceso extenso tomar la información necesaria (extensión, servidor, fecha) e importarla a una base de datos fácilmente, organizar esta información de forma manual era una tarea demasiado larga dado que cada documento contaba con aproximadamente 30 hojas de datos cada una con 5000 registros y estos reportes se venían realizando semanalmente desde diciembre del 2015.

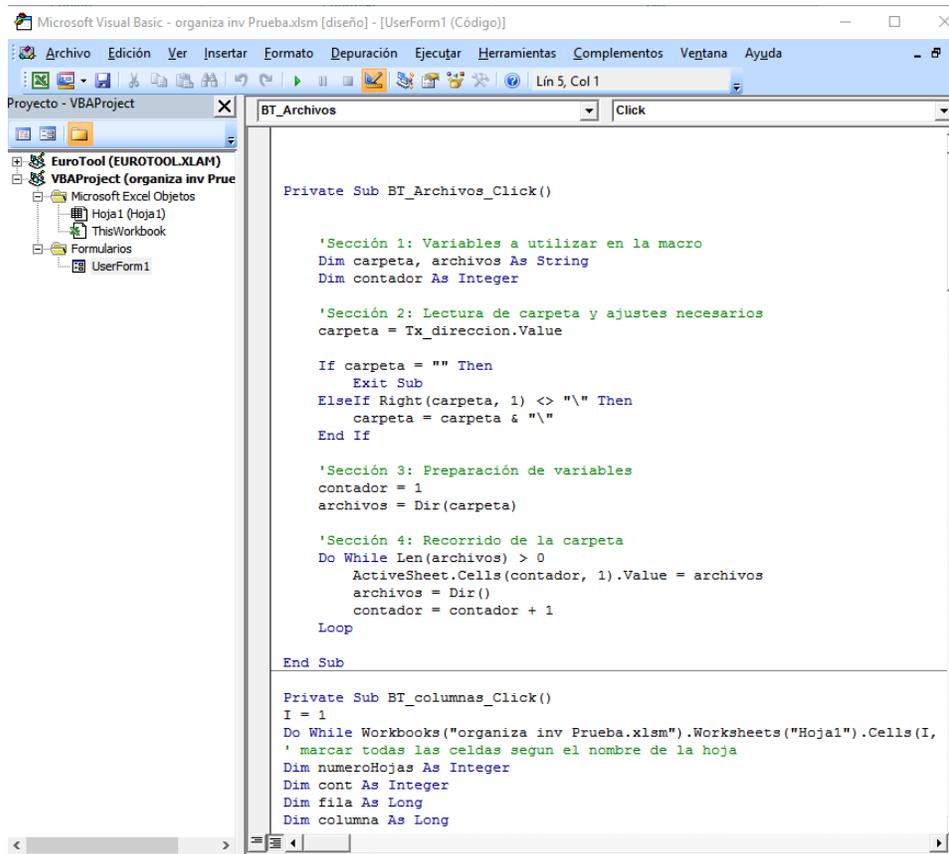
Figura 4. Visualización documentos antiguos.



 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

con el fin de resolver esta problemática, se opta por la creación de un software desarrollado en el compilador de Excel el cual se muestra en la Figura 5, el cual utiliza el lenguaje Visual Basic 6.0. Se decide utilizar este compilador dado que presenta librerías y funciones únicas que permiten la manipulación de los documentos realizados en el mismo.

Figura 5. Visualización Entorno compilador Excel.



```

Microsoft Visual Basic - organiza inv Prueba.xlsm [diseño] - [UserForm1 (Código)]
Archivo Edición Ver Insertar Formato Depuración Ejecutar Herramientas Complementos Ventana Ayuda
Proyecto - VBAProject
EuroTool (EUROTOOL.XLAM)
VBAProject (organiza inv Prue
  Microsoft Excel Objetos
  Hoja1 (Hoja1)
  ThisWorkbook
  Formularios
  UserForm1
BT_Archivos Click
Private Sub BT_Archivos_Click()

'Sección 1: Variables a utilizar en la macro
Dim carpeta, archivos As String
Dim contador As Integer

'Sección 2: Lectura de carpeta y ajustes necesarios
carpeta = Tx_direccion.Value

If carpeta = "" Then
Exit Sub
ElseIf Right(carpeta, 1) <> "\" Then
carpeta = carpeta & "\"
End If

'Sección 3: Preparación de variables
contador = 1
archivos = Dir(carpeta)

'Sección 4: Recorrido de la carpeta
Do While Len(archivos) > 0
ActiveSheet.Cells(contador, 1).Value = archivos
archivos = Dir()
contador = contador + 1
Loop

End Sub

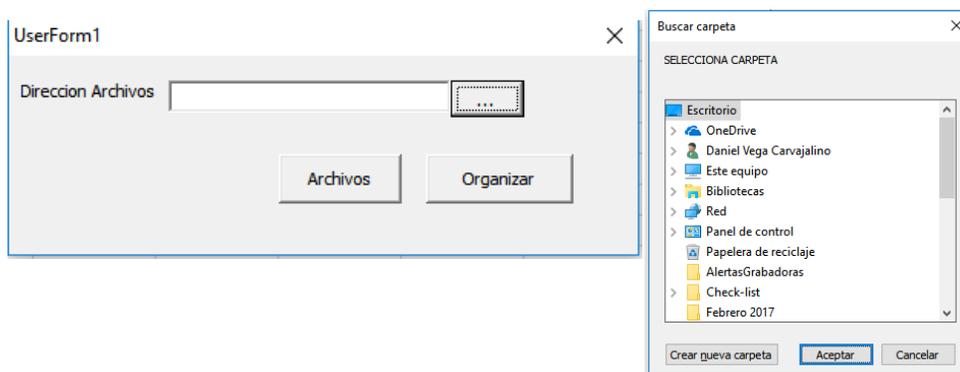
Private Sub BT_columnas_Click()
I = 1
Do While Workbooks("organiza inv Prueba.xlsm").Worksheets("Hoja1").Cells(I,
' marcar todas las celdas segun el nombre de la hoja
Dim numeroHojas As Integer
Dim cont As Integer
Dim fila As Long
Dim columna As Long

```

El programa propuesto consta de un formulario (Figura 6) en el cual se ingresa la ruta de los documentos a modificar, detecta el nombre de los archivos en la carpeta, los abre, modifica y guarda ya estandarizados en el formato que sera importado posteriormente por la base de datos

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Figura 6. Primera Pantalla software Parametrización.

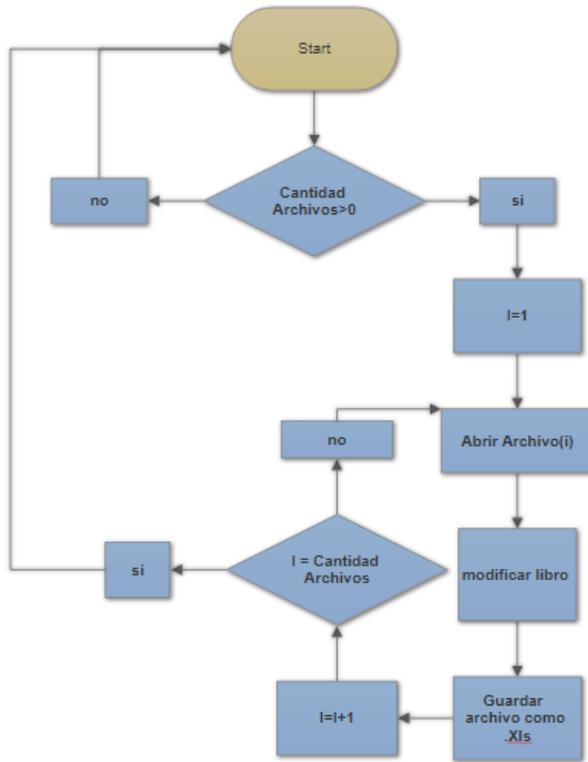


El botón “archivos” se encarga de abrir la ruta seleccionada anteriormente, buscar todos los archivos de Excel en ella y guardar los nombres para posteriormente ser modificados.

El botón “organizar” tiene como función leer los datos almacenadas por “archivos” y entrar al documento modificar cada hoja eliminando todos los datos innecesarios, detectar si en la hoja existen extensiones duplicadas para borrarlas y no guardar información redundante. Posteriormente lee el nombre de la hoja el cual contiene la IP del servidor donde se encuentra dicha extensión y se lo asigna los datos, de igual forma detecta la fecha en la que se creó el archivo asignándole esta información a cada registro para que al momento de importar la tabla SQL se conozca en que servidor se encontraba la extensión para esa fecha. Finalmente se copia toda la información en una sola hoja y se eliminan las demás dejando toda la información en un solo espacio del documento, esta se guarda como un archivo .XLS el cual es el más utilizado a la hora de importar datos a una base de datos.

Este proceso se representa mediante el diagrama de flujo en la figura 7 donde se muestra la lógica utilizada en esta fase del proyecto.

Figura 7. Diagrama representativo del software.



Una vez el software termina el proceso en todos los documentos estos están listos para ser importados a la base de datos. Según el estándar deseado el cual se puede ver en la figura 8.

Figura 8. Información organizada y parametrizada.

Extension	Grabadora	Fecha
69116	172.20.138.203	10/2/2017
69151	172.20.138.203	10/2/2017
69207	172.20.138.203	10/2/2017
69208	172.20.138.203	10/2/2017
69244	172.20.138.203	10/2/2017
69254	172.20.138.203	10/2/2017
82171	172.20.138.203	10/2/2017
82172	172.20.138.203	10/2/2017
82173	172.20.138.203	10/2/2017
82174	172.20.138.203	10/2/2017
82690	172.20.138.203	10/2/2017
82684	172.20.138.203	10/2/2017
82685	172.20.138.203	10/2/2017

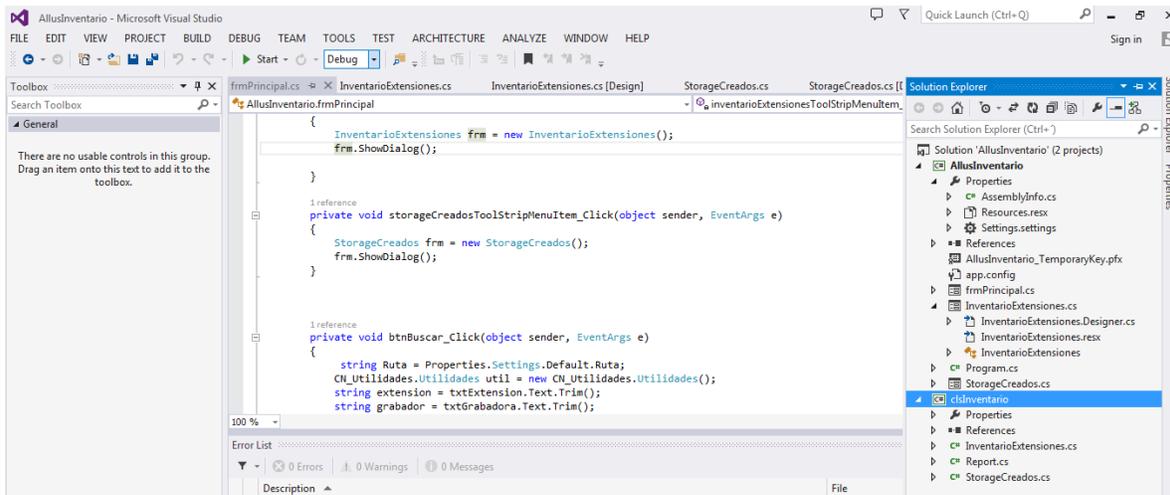
 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3.6 SOFTWARE DE AUTOMATIZACIÓN.

El software se desarrolló en Visual Studio 2013 y utilizo el lenguaje C# como se muestra en la figura 9, el cual se consideró el más adecuado para el proyecto por su compatibilidad en entornos Windows y sus herramientas para vincular el software a SQL; este se encargara de importar los datos estandarizados a las bases de datos, estos datos no son únicamente los inventarios que fueron organizados en el software de parametrización sino que también será el encargado de importar los reportes que contienen la información de las rutas donde cada uno de los servidores se encontraba almacenando para una fecha específica.

El software es capaz de cruzar la información de estos documentos para de realizar automáticamente la búsqueda de las llamadas en una fecha específica.

Figura 9. Visualización Visual Studio 2013



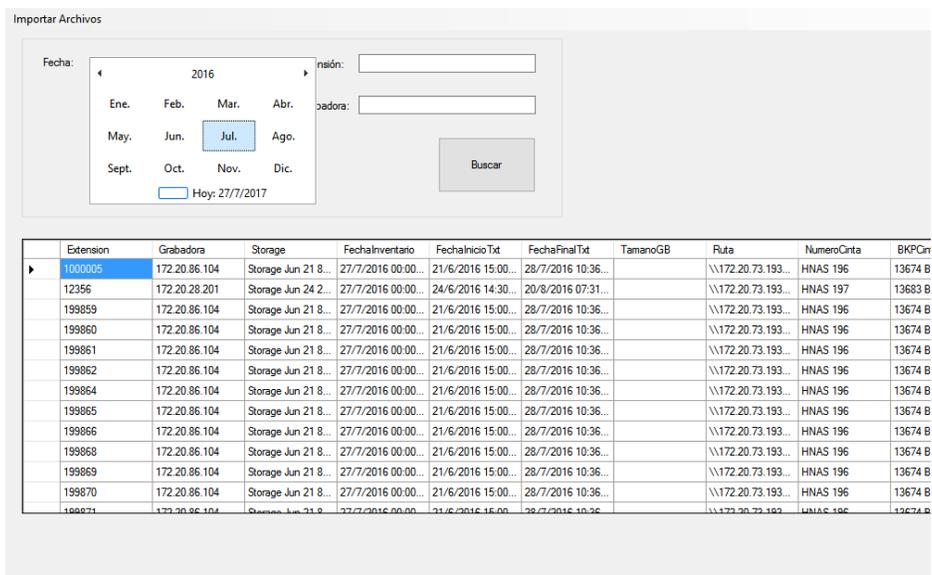
Este programa cuenta con 2 métodos, un módulo operativo en el cual se realizan las búsquedas de la información y el un módulo de alimentación de las bases de datos.

3.6.1 Búsqueda y visualización de la información.

Esta interfaz cuenta con un calendario interactivo en el cual el usuario puede realizar la solicitud de la llamada. Se optó por hacerlo gráfico en vez de escrito dado que una fecha se puede escribir de diferentes maneras dependiendo del usuario sin embargo de esta forma el usuario selecciona la fecha deseada y el software convierte la misma a una forma numérica que es más precisa a la hora de buscar en las bases de datos.

También cuenta con 2 tipos de consulta, una por extensión la cual muestra en que servidor se encontraba la misma para esta fecha y a su vez en que storage se encontraba grabando. Adicionalmente se puede realizar dicha búsqueda por la IP del servidor (grabadora) y este mostrara el storage del mismo para la fecha y todas las extensiones asociadas a él. Como se muestra en la figura 10.

Figura 10. Interfaz de Búsqueda y visualización



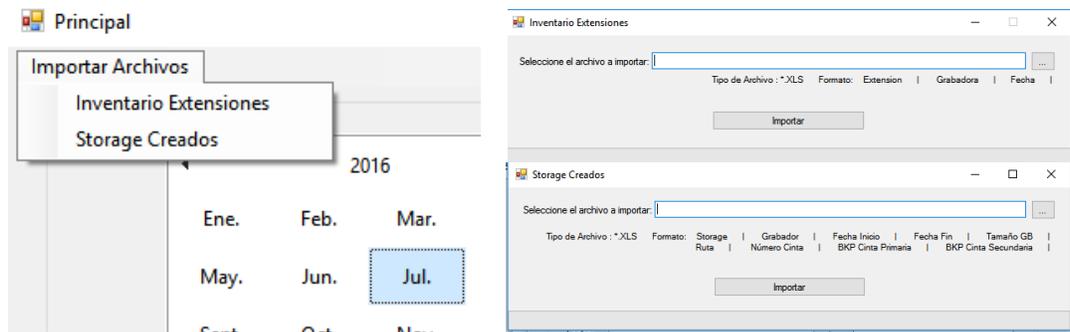
The screenshot shows a web interface for searching records. At the top, there is a section titled 'Importar Archivos' with a date selector for 2016, currently showing July. Below the calendar are input fields for 'Extensión', 'Grabadora', and 'Storage', and a 'Buscar' button. Below the search form is a table with the following columns: Extension, Grabadora, Storage, FechaInventario, FechaInicioTxt, FechaFinalTxt, TamanoGB, Ruta, NumeroCrita, and BKPCri. The table contains 17 rows of data, with the first row highlighted in blue.

Extension	Grabadora	Storage	FechaInventario	FechaInicioTxt	FechaFinalTxt	TamanoGB	Ruta	NumeroCrita	BKPCri
1000005	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
12356	172.20.28.201	Storage Jun 24 2...	27/7/2016 00.00...	24/6/2016 14.30...	20/8/2016 07:31...		\\172.20.73.193...	HNAS 197	13683 B
199859	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199860	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199861	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199862	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199864	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199865	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199866	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199868	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199869	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199870	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B
199871	172.20.86.104	Storage Jun 21 8...	27/7/2016 00.00...	21/6/2016 15.00...	28/7/2016 10:36...		\\172.20.73.193...	HNAS 196	13674 B

3.6.2 Almacenamiento y Organización de los archivos en las bases de datos.

En este módulo del software se buscan y seleccionan inventarios y reportes más recientes para mantener actualizada la información en la base de datos. A la hora de importar los datos el software indica cómo debe estar dispuesta la información en el documento para que esta pueda almacenarse en la base de datos y evitar que a esta le lleguen registros erróneos como se muestra en la figura 11.

Figura 11. Almacenamiento y Organización



No obstante, en caso de que el documento no se encuentre bien diseñado el programa es capaz de detectar esta falla y no importará dicha información, además de generar un archivo .txt el cual señala los errores encontrados y muestra la línea de la tabla que no pudo ser importada (Figura 12). Para que el usuario pueda corregir la información pertinente, además el sistema detecta las líneas repetidas y actualiza el registro faltante sin repetir el existente.

Figura 12. Log creado cuando encuentra una falla en la tabla

```

# Fila ; Detalle del Error
5 ; Fecha Inicial Invalida; Fecha Final Invalida;
6 ; Fecha Inicial Invalida; Fecha Final Invalida;
7 ; Fecha Inicial Invalida; Fecha Final Invalida;
8 ; Fecha Inicial Invalida; Fecha Final Invalida;
10 ; Fecha Inicial Invalida; Fecha Final Invalida;

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

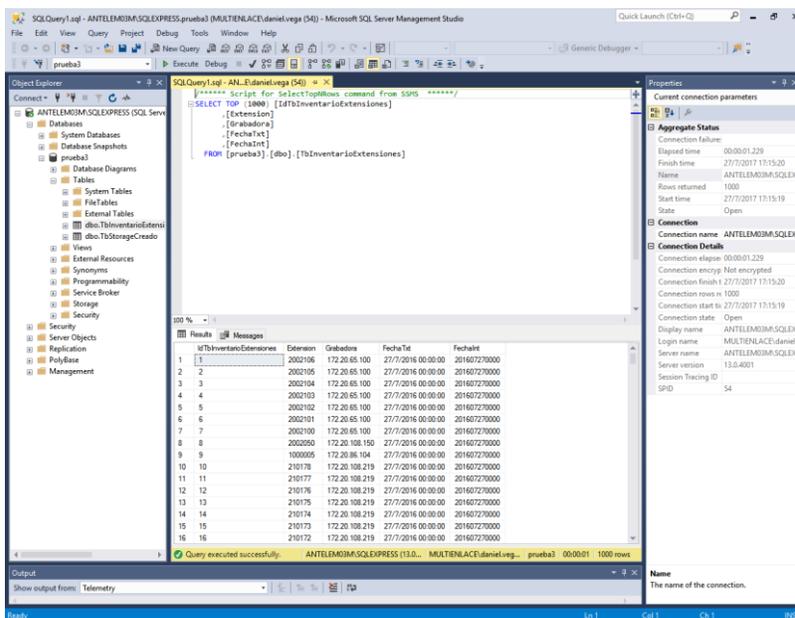
3.7 Creación base de datos

Para la creación de las bases de datos se utilizó SQL Management Studio Express (Figura 13) dado su buen funcionamiento, fácil parametrización, una vinculación simple de realizar al programa creado y su uso no depende de una licencia. Esto limita un poco la funcionabilidad de la base en cuanto a características que no se necesitan para la aplicación actual por lo tanto no es necesario adquirir licencias con más funcionalidades ya que este se adapta a nuestras necesidades de diseño.

La base de datos cuenta con 2 tablas, en una se almacenan todos los inventarios y en la otra el reporte de storage existentes. Una vez esta reciba la solicitud del programa, cruzara los datos de las 2 tablas y envia la información de vuelta para que esta sea visualizada en el software.

Para mejorar el desempeño de la base de la búsqueda y mejorar los tiempos de respuesta, se ha desarrollado un procedimiento almacenado que se invoca directamente de la aplicación así el procesamiento lo realiza el motor de la base de datos directamente.

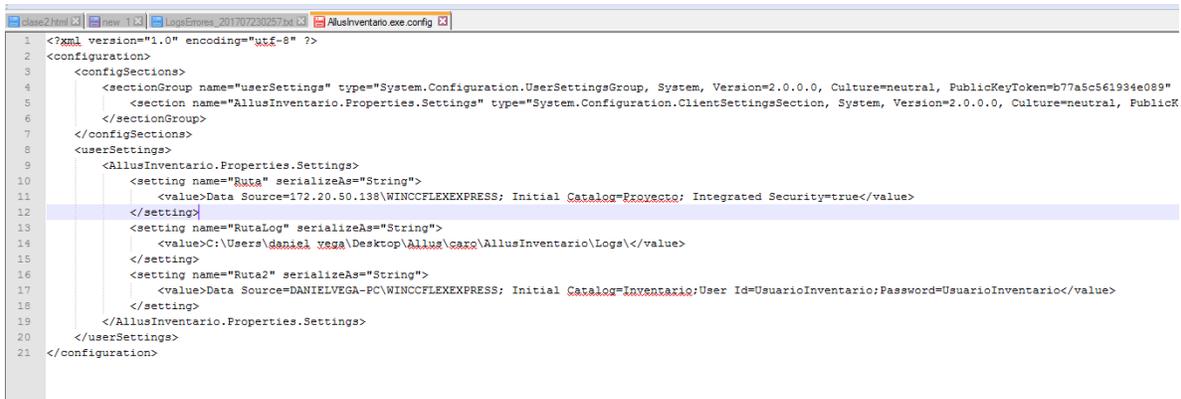
Figura 13. Entorno SQL Express



	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para facilitar el proceso de configuración y vinculación de la base y el programa se parametrizo utilizando el archivo de configuración .NET y en él se estableció la ubicación de la base y el nombre de la misma. En esta configuración también es posible establecer la ruta donde se almacenarán los archivos que contendrán los errores generados por el programa (logs) tal como se muestra en la figura 14.

Figura 14. Formulario de parametrización



```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <configuration>
3   <configSections>
4     <sectionGroup name="UserSettings" type="System.Configuration.UserSettingsGroup, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
5     <section name="AllusInventario.Properties.Settings" type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral, PublicK
6     </sectionGroup>
7   </configSections>
8   <userSettings>
9     <AllusInventario.Properties.Settings>
10      <setting name="Ruta" serializeAs="String">
11        <value>Data Source=172.20.50.138\WINCCFLEXEXPRESS; Initial Catalog=Proyecto; Integrated Security=true</value>
12      </setting>
13      <setting name="RutaLog" serializeAs="String">
14        <value>C:\Users\daniel.vega\Desktop\Allus\caro\AllusInventario\Logs\</value>
15      </setting>
16      <setting name="Ruta2" serializeAs="String">
17        <value>Data Source=DANIELVEGA-PC\WINCCFLEXEXPRESS; Initial Catalog=Inventario; User Id=UsuarioInventario; Password=UsuarioInventario</value>
18      </setting>
19    </AllusInventario.Properties.Settings>
20  </userSettings>
21 </configuration>

```

	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. RESULTADOS Y DISCUSIÓN

El proyecto logra automatizar el proceso de búsqueda y recuperación de las llamadas de la empresa Konecta, creando un nuevo estándar en el almacenamiento de información y diseñando un programa capaz de realizar dicha labor de forma automática, no obstante al ser un nuevo software dentro de la empresa algunos servidores bloquearon su funcionamiento, por lo que fue necesaria la intervención del área de seguridad y desarrollo las cuales tuvieron que analizar el mismo, para validar que al darle los permisos necesarios este no interfiriera con otros procesos dentro de la empresa, motivo por el cual su salida a producción se retrasó y tuvo que ser instalado en un servidor diferente para que todos los computadores en la red tuvieran acceso a la base de datos y al programa.

Por lo tanto, se puede concluir que el software cumple con los objetivos planteados y con todos los requerimientos de seguridad necesarios para operar reduciendo el tiempo de búsqueda y recuperación en un 50%, mejorando considerablemente la efectividad de área.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

- Se automatizó el proceso de búsqueda y recuperación de llamadas aumentando la eficiencia en el área.
- Se cumplió el objetivo del primer software el cual era unificar y estandarizar los archivos más antiguos los cuales eran creados de forma manual.
- Se creó un nuevo estándar en el cual se almacenarán los inventarios y se logró reducir la intervención humana en el proceso de creación de las mismas.
- Gracias al apoyo de las áreas de desarrollo y seguridad se logró solucionar todos los problemas que se presentaron durante el proceso de desarrollo e implementación del sistema.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

- SYBEX Andrew G. Blank. (2004). TCP/IP Foundations.
- Mercedes Marqués Andrés (2006) Iniciación a la programación en C#: un enfoque práctico.
- Yolanda Cerezo López, Olga Peñalba (2001) SQL y desarrollo de aplicaciones en Oracle 8.
- Microsoft (2017), Definition Visual Basic, <https://docs.microsoft.com/en-us/dotnet/visual-basic/index>
- Autómatas Programables (2001), La automatización, <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMh1/PAGINA%20PRINCIPAL/Automatizacion/Automatizacion.htm>

APÉNDICE

A. Programación VBA.

El software fue desarrollado en el compilador de Excel dado que este cuenta con funciones propias que permite la manipulación sencilla de los documentos de este tipo.

1. A partir de una hoja de Excel se crea un botón (figura 15) el cual se enlaza a un formulario (figura 16) el cual cuenta con el programa que se encargara de organizar y parametrizar los archivos.

Figura 15. Boton que activa el Formulario

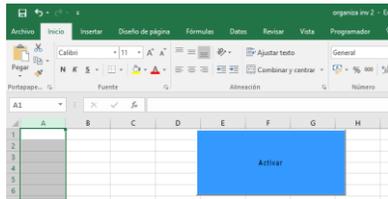
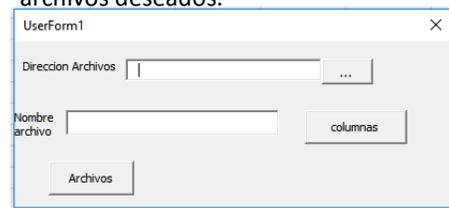


Fig.16 Formulario que accede y modifica los archivos deseados.



2. Códigos

- 2.1 Botón "...": se encarga de abrir un explorador de Windows y guardar la ruta del archivo.

```

Private Sub Buscar_Click()
Set navegador = CreateObject("shell.application")carpeta =
navegador.browseforfolder(0, "SELECCIONA CARPETA", 0,
""),items.Item.Path
ChDir carpeta & "\"
Tx_direccion.Value = carpeta
End Sub

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.2 Botón “Archivos”: Se encarga de leer y guardar en una lista el nombre de los archivos a modificar.

```

Private Sub BT_Archivos_Click(

    Dim carpeta, archivos As String

    Dim contador As Integer

    carpeta = Tx_direccion.Value

    If carpeta = "" Then

        Exit Sub

    ElseIf Right(carpeta, 1) <> "\" Then

        carpeta = carpeta & "\"

    End If

    contador = 1

    archivos = Dir(carpeta)

    Do While Len(archivos) > 0

        ActiveSheet.Cells(contador, 1).Value = archivos

        archivos = Dir()

        contador = contador + 1

    Loop

End Sub

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.3 Botón “columnas” se encarga de abrir y organizar todos los elementos del archivo es el botón principal del programa.

```

Private Sub BT_columnas_Click()

I = 1

Do While Workbooks("organiza inv
2").Worksheets("Hoja1").Cells(I, 1).Value <> ""

' marcar todas las celdas segun el nombre de la hoja

Dim numeroHojas As Integer

Dim cont As Integer

Dim fila As Long

Dim columna As Long

Dim x As Integer

If Tx_direccion.Value = "" Or TX_nombre = "" Then

MsgBox "favor digitar direccion o nombre valido"

Exit Sub

End If

'Indicar el libro de Excel destino

Set wbDestino =
Workbooks.Open(Tx_direccion.Value & "\" &
Workbooks("organiza inv
2").Worksheets("Hoja1").Cells(I, 1).Value)

'Activar este libro

wbDestino.Activate

```

```

Application.DisplayAlerts = False

Sheets(cont).Columns("A:A").TextToColumns _
Destination:=Range("A1"), _
DataType:=xlDelimited, _
ConsecutiveDelimiter:=True, _
Other:=True, _
OtherChar:="("

Application.DisplayAlerts = True

Sheets(cont).Columns("B:B").Delete

Sheets(cont).Columns("A:A").TextToColumns _
Destination:=Range("A1"), _
DataType:=xlDelimited, _
ConsecutiveDelimiter:=True, _
Other:=True, _
OtherChar:="-"

Application.DisplayAlerts = True

Sheets(cont).Columns("B:B").Delete

Sheets(cont).Columns("A:A").RemoveDuplicates
Columns:=1, Header:=xlYes

Next cont

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
For cont = 1 To numeroHojas
```

```
NumRows = 0
```

```
' Establecer numrows = número de filas de datos.
```

```
Sheets(cont).Select
```

```
NumRows = Sheets(cont).Range("A1",  
Sheets(cont).Range("A1").End(xlDown)).Rows.Count
```

```
' Seleccionar celda a1.
```

```
Sheets(cont).Range("A1").Select
```

```
' Establecer valor "For" para que vaya de bucle el  
bucle durante "numrows" veces.
```

```
For x = 1 To NumRows
```

```
' Inserte el código aquí.
```

```
Worksheets(cont).Cells(x, 2).Value =  
Sheets(cont).Name
```

```
' Selecciona la celda 1 fila por debajo de la celda  
activa.
```

```
ActiveCell.Offset(1, 0).Select
```

```
Next
```

```
Application.DisplayAlerts = False
```

```
For Each hoja In ActiveWorkbook.Sheets
```

```
If hoja.Name = "todas" Then hoja.Delete
```

```
Next
```

```
Sheets.Add before:=Sheets(1)
```

```
ActiveSheet.Name = "todas"
```

```
For x = 2 To Sheets.Count
```

```
Sheets(x).Select
```

```
Range("a2:o" &  
Range("a65000").End(xlUp).Row).Copy
```

```
Sheets("todas").Range("a65000").End(xlUp).Offset(1,  
0).PasteSpecial Paste:=xlValues
```

```
Next
```

```
Sheets("todas").Select
```

```
Dim ws As Worksheet
```

```
For Each ws In ActiveWorkbook.Worksheets
```

```
If ws.Name <> ActiveWorkbook.ActiveSheet.Name  
Then
```

```
Application.DisplayAlerts = False
```

```
ws.Delete
```

```
Application.DisplayAlerts = True
```

```
End If
```

```
Next ws
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

fecha = Replace(ActiveWorkbook.Name, ".xlsx", "")
fecha = Replace(fecha, "inventario Medellin", "")

x = 2

Do While Worksheets("todas").Cells(x, 1).Value <> "

    ini = Left(Worksheets("todas").Cells(x, 1).Value, 1)

    If ini <> "1" And ini <> "2" And ini <> "3" And ini <> "4" And
ini <> "5" And ini <> "6" And ini <> "7" And ini <> "8" And ini <> "9"
And ini <> "0" Then

        Rows(x).EntireRow.Delete

    Else

        Worksheets("todas").Cells(x, 3).Value = fecha

    x = x + 1

    End If

Loop

ActiveWorkbook.Save

ActiveWorkbook.Close

l = l + 1

Loop

End Sub

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

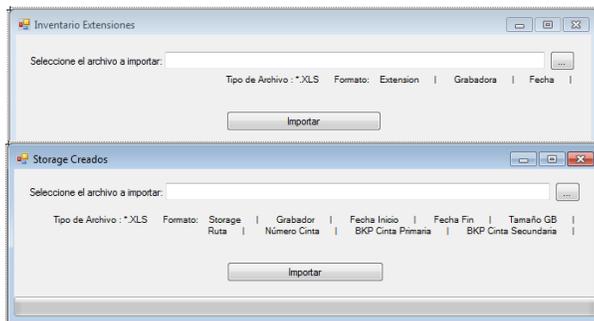
B. Programación C#.

La programación en este lenguaje se divide en 2 métodos uno encargado de importar los archivos de Excel a las bases de datos y otro encargado de realizar el cruce de información y filtrar los datos que se desean visualizar.

Es necesario aclarar que para la facilidad y eficiencia de la programación se utilizaron librerías y clases externas al software.

1 importar información.

En él se puede importar la información de los inventarios y reportes cuya información se desea importar.



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace AllusInventario
{
    public partial class
    InventarioExtensiones : Form
    {
        public InventarioExtensiones()
        {
            InitializeComponent();
        }

        private void
        btnSelect_Click(object sender, EventArgs

```

```

using System;
e)
    {
        this.openFileDialog1.Filter =
        "XLS Files (*.XLS)|*.XLS";
        openFileDialog1.ShowDialog();
        if
        (openFileDialog1.CheckFileExists)
        {
            txtPath.Text =
            openFileDialog1.FileName;
        }
        private void
        btnImportar_Click(object sender, EventArgs
        e)
        {
            if
            (System.IO.File.Exists(txtPath.Text))
            {

```

```

sing System;
e)
    {
        this.openFileDialog1.Filter
= "XLS Files (*.XLS)|*.XLS";
openFileDialog1.ShowDialog();
        if
(openFileDialog1.CheckFileExists)
        {
            txtPath.Text =
openFileDialog1.FileName;
        }
    }
    private void
btnImportar_Click(object sender,
EventArgs e)
    {
        if
(System.IO.File.Exists(txtPath.Text))
        {
            progressBar1.Value = 0;
            progressBar1.Step = 1;
            string Ruta =
Properties.Settings.Default.Ruta;
            CapaDatos.DatosExcel
vExcel = new CapaDatos.DatosExcel();

            DataSet DS;

            try
            {
                DS =
vExcel.TraeHojaDeExcel(txtPath.Text,
"todas");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message.ToString(),
"Error leyendo el archivo",
MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return;
            }

            if (DS.Tables.Count ==
0)
            {
                MessageBox.Show("El
libro seleccionado no corresponde al

```

```

formato esperado", "Datos Incorrectos",
MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return;
            }
            if
(DS.Tables[0].Rows.Count < 2)
            {
                MessageBox.Show("No
se han encontrado datos en el libro
seleccionado", "Datos Incorrectos",
MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return;
            }

            int Conteo = 0;
            progressBar1.Maximum =
DS.Tables[0].Rows.Count;

            CN_Utilidades.Utilidades util = new
CN_Utilidades.Utilidades();
            StringBuilder SBErrores
= new StringBuilder();
            SBErrores.AppendLine("#
Fila ; Detalle del Error");

            foreach (DataRow dr in
DS.Tables[0].Rows)
            {
                string TextError =
"";

                if (Conteo > 0)
                {
                    DateTime dt;
                    if
(DateTime.TryParse(dr[2].ToString(),
out dt) == false)
                    {
                        TextError =
TextError + "Fecha Invalida; ";
                    }

                    if (TextError
== "")
                    {
                        try
                        {

```

```

clsInventario.InventarioExtensiones
extension = new
clsInventario.InventarioExtensiones();

extension.Extension = dr[0].ToString();
extension.Grabadora = dr[1].ToString();
extension.FechaTxt = dr[2].ToString();

extension.FechaInt =
util.FechaHoraInt(dt);

                                int rta
=
extension.InsertaInventarioExtensiones(
Ruta, extension);
                                }
                                catch
(Exception ex)
                                {
TextError = TextError + ex.Message + ";
";
                                }
                                if
(TextError != "")
                                {
SBErrores.AppendLine(Conteo.ToString()
+ " ; " + TextError);
                                }
                                }
                                else
                                {
                                if
(dr.ItemArray.Length < 3)
                                {
MessageBox.Show("EL archivo
seleccionado no tiene el formato
esperado", "Datos Incorrectos",
MessageBoxButtons.OK,
MessageBoxIcon.Error);
                                return;
                                }
                                }
Conteo += 1;

```

```

                                Conteo += 1;
progressBar1.PerformStep();
                                }
                                string ConErrores
= "";
                                if (SBErrores.Length >
1)
                                {
                                string
NombreArchivo
=Properties.Settings.Default.RutaLog +
"Errores_" +
util.FechaHoraInt(DateTime.Now).ToStrin
g() ;
System.IO.File.WriteAllText(NombreArchi
vo, SBErrores.ToString());
                                ConErrores = ". No
se han importado " + SBErrores.Length +
" líneas, puede revisar el detalle en
el archivo de log: " + NombreArchivo;
                                }
                                MessageBox.Show("Carga
Finalizada", "Datos Importados",
MessageBoxButtons.OK,
MessageBoxIcon.Information);
                                }
                                }
                                }
}

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

C. Configuración SQL.

```

USE [Proyecto]

GO

/***** Object: StoredProcedure
[dbo].[PA_BuscaGrabaciones]  Script Date:
23/07/2017 9:04:32 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>

CREATE PROCEDURE [dbo].[PA_BuscaGrabaciones]

    -- Add the parameters for the stored
    procedure here

    @Extension nvarchar(50)

    ,@Grabador nvarchar(50)

    ,@FechaInicial money

    ,@FechaFinal money

AS

```

```

BEGIN

    -- SET NOCOUNT ON added to prevent extra
    result sets from

    -- interfering with SELECT statements.

    SET NOCOUNT ON;

    declare @Conteo int;

    set @Conteo = 0;

    select @Conteo = count(*) from
    TbInventarioExtensiones AS inv where inv.FechaInt
    =@FechaInicial;

    if @Conteo > 0 and @FechaFinal = 0 begin

        SELECT distinct  inv.Extension,
        inv.Grabadora, Storage, inv.FechaTxt
        FechaInventario, sto.FechaInicioTxt,
        sto.FechaFinalTxt, sto.TamanoGB, sto.Ruta,
        sto.NumeroCinta, sto.BKPCintaPrimaria,
        sto.BKPCintaSecundaria--, sto.IdTbStorageCreado

        FROM

        TbInventarioExtensiones AS inv INNER JOIN

        TbStorageCreado AS sto ON
        inv.Grabadora = sto.Grabadora

```

 ITM Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

AND inv.FechaInt >= sto.FechaInicioInt AND
inv.FechaInt <= sto.FechaFinalInt

      where ( inv.Extension = @Extension
or @Extension = ")

      AND ( inv.Grabadora = @Grabador
or @Grabador = ")

      AND ( inv.FechaInt = @FechaIncial)

      END else Begin

      SELECT distinct inv.Extension,
inv.Grabadora, Storage, inv.FechaTxt
FechaInventario, sto.FechaInicioTxt,
sto.FechaFinalTxt, sto.TamanoGB, sto.Ruta,
sto.NumeroCinta, sto.BKPCintaPrimaria,
sto.BKPCintaSecundaria--, sto.IdTbStorageCreado

      FROM
TbInventarioExtensiones AS inv INNER JOIN

      TbStorageCreado AS sto ON
inv.Grabadora = sto.Grabadora AND inv.FechaInt >=
sto.FechaInicioInt AND inv.FechaInt <=
sto.FechaFinalInt

      where ( inv.Extension = @Extension
or @Extension = ")

      AND ( inv.Grabadora = @Grabador
or @Grabador = ")

      AND ( inv.FechaInt >=
@FechaIncial or @FechaIncial = 0)

      AND ( inv.FechaInt <= @FechaFinal
or @FechaFinal = 0)

```

```

GO

/***** Object: StoredProcedure
[dbo].[PA_InventarioExtensiones_Insertar] Script
Date: 23/07/2017 9:04:32 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

--
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- CREATE PROCEDURE
[dbo].[PA_InventarioExtensiones_Insertar]

      -- Add the parameters for the stored
procedure here

      @Extension nvarchar(50)
      ,@Grabadora nvarchar(50)
      ,@FechaTxt nvarchar(50)
      ,@FechaInt money
      ,@Resultado nvarchar(50) output

AS

BEGIN

      -- SET NOCOUNT ON added to prevent extra

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

-- SET NOCOUNT ON added to prevent extra
result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON

Declare @Id int

set @Id = 0;

select @Id =
isnull(IdTbInventarioExtensiones,0) from
TbInventarioExtensiones

where Extension = @Extension and
Grabadora = @Grabadora and FechaTxt =
@FechaTxt;

set @Resultado = 0;

if @Id = 0 BEGIN

INSERT INTO TbInventarioExtensiones
(Extension,
Grabadora, FechaTxt, FechaInt)

VALUES
(@Extension,@Grabadora,@FechaTxt,@FechaInt);

set @Resultado =@@rowcount;

end

return @Resultado;

END

GO

[dbo].[PA_StorageCreados_Insertar] Script Date:
23/07/2017 9:04:32 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```

```

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

CREATE PROCEDURE
[dbo].[PA_StorageCreados_Insertar]

-- Add the parameters for the stored
procedure here

@Storage nvarchar(50)

,@Grabadora nvarchar(50)

,@FechaInicioTxt nvarchar(50)

,@FechaInicioInt money

,@FechaFinalTxt nvarchar(50)

,@FechaFinalInt money

,@TamanoGB nvarchar(50)

,@Ruta nvarchar(max)

,@NumeroCinta nvarchar(50)

,@BKPCintaPrimaria nvarchar(max)

,@BKPCintaSecundaria nvarchar(max)

,@Resultado nvarchar(50) output

AS

BEGIN -- SET NOCOUNT ON added to prevent extra
result sets from

-- interfering with SELECT statements.

Declare @Id int

set @Id = 0;

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

select @Id = isnull(IdTbStorageCreado,0)
from TbStorageCreado

```

```

where Storage = @Storage and
FechalnicioTxt = @FechalnicioTxt and FechaFinalTxt
= @FechaFinalTxt;

```

```

set @Resultado = 0;

```

```

if @Id = 0 BEGIN

```

```

    INSERT INTO TbStorageCreado

```

```

        (Storage,Grabadora, FechalnicioTxt,
FechalnicioInt, FechaFinalTxt, FechaFinalInt,
TamanoGB, Ruta, NumeroCinta, BKPCintaPrimaria,
BKPCintaSecundaria)

```

```

        VALUES

```

```

        (@Storage,@Grabadora,@FechalnicioTxt,@Fechaln
icioInt,@FechaFinalTxt,@FechaFinalInt,@TamanoG
B,@Ruta,@NumeroCinta,@BKPCintaPrimaria,@BKP
CintaSecundaria)

```

```

        set @Resultado =@@rowcount;

```

```

    end else BEGIN

```

```

        UPDATE    TbStorageCreado

```

```

        SET      TamanoGB =@TamanoGB,
Ruta =@Ruta, NumeroCinta =@NumeroCinta,
BKPCintaPrimaria =@BKPCintaPrimaria,
BKPCintaSecundaria =@BKPCintaSecundaria

```

```

        where Storage = @Storage and
FechalnicioTxt = @FechalnicioTxt and FechaFinalTxt
= @FechaFinalTxt;

```

```

    end

```

```

return @Resultado;

```

```

END

```

```

GO

```

```

[dbo].[TbInventarioExtensiones]  Script Date:
23/07/2017 9:04:32 *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

CREATE TABLE [dbo].[TbInventarioExtensiones](

```

```

    [IdTbInventarioExtensiones] [int]
IDENTITY(1,1) NOT NULL,

```

```

    [Extension] [nvarchar](50) NULL,

```

```

    [Grabadora] [nvarchar](50) NULL,

```

```

    [FechaTxt] [nvarchar](50) NULL,

```

```

    [FechaInt] [numeric](18, 0) NULL,

```

```

    CONSTRAINT [PK_TbInventarioExtensiones]
PRIMARY KEY CLUSTERED

```

```

(

```

```

    [IdTbInventarioExtensiones] ASC

```

```

)WITH (PAD_INDEX = OFF,

```

```

STATISTICS_NORECOMPUTE = OFF,

```

```

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```

) ON [PRIMARY]

```

```

GO

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para el proceso de configuración y vinculación de la base y el programa se parametrizo

```

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

  <configSections>

    <sectionGroup name="userSettings" type="System.Configuration.UserSettingsGroup, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" >

      <section name="AllusInventario.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" allowExeDefinition="MachineToLocalUser" requirePermission="false"
/>

    </sectionGroup>
  </configSections>

  <userSettings>

    <AllusInventario.Properties.Settings>

      <setting name="Ruta" serializeAs="String">

        <value>Data Source=ANTELEM03M\SQLEXPRESS; Initial Catalog=Inventario; Integrated
Security=true</value>

      </setting>

      <setting name="RutaLog" serializeAs="String">

        <value>C:\proyecto\LOGS</value>

      </setting>

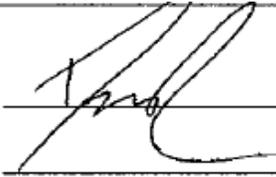
    </AllusInventario.Properties.Settings>

  </userSettings>

</configuration>

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES 

FIRMA ASESOR Julliana Valencia A

Informe final de práctica profesional.

FECHA ENTREGA: 10/08/2017

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO ___ ACEPTADO ___ ACEPTADO CON MODIFICACIONES ___

ACTA NO. _____

FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____

ACTA NO. _____

FECHA ENTREGA: _____