



UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
VERITATE SOLA NOBIS IMPONETUR VIRILISTOGA. 1948

Facultad de Ingeniería Eléctrica

**Centro de Estudios de Electrónica y Tecnologías de
la Información**



TRABAJO DE DIPLOMA

**Sistema Automatizado para el Control de Activos
Fijos Tangibles**

Autor: Enrique San Gil Hernández

Tutores: Ing. Alexander Falcón Ruiz

Ing. Denis Hernández Pacheco

Santa Clara, Villa Clara

2008

"Año 50 de la Revolución."



Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

**Centro de Estudios de Electrónica y Tecnologías de la
Información (CEETI)**



TRABAJO DE DIPLOMA

Sistema Automatizado para el Control de Activos

Fijos Tangibles

Autor: Enrique San Gil Hernández.

enrique@uclv.edu.cu

Tutores: Ing. Alexander Falcón Ruiz.

Profesor Asistente, CEETI,

Facultad de Ingeniería Eléctrica, UCLV.

afalcon@uclv.edu.cu

Ing. Denis Hernández Pacheco.

Instructor en adiestramiento, CEETI

Facultad de Ingeniería Eléctrica, UCLV.

hpacheco@uclv.edu.cu



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Telecomunicaciones y Electrónica, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Jefe de Departamento
donde se defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

PENSAMIENTO

“Es de importancia para quien desee alcanzar una certeza en su investigación, el
saber dudar a tiempo”

Aristóteles

DEDICATORIA

A mis padres, por haber depositado en mi toda su confianza y haberme guiado por el
camino correcto.

A mis amigos, por estar juntos hasta el final del camino.

A mi hermano y toda mi familia.

AGRADECIMIENTOS

A mis tutores Alexander y Denis, por ofrecerme su valiosa ayuda para llegar a concluir este trabajo.

A mis profesores, por transmitirme los conocimientos necesarios para llegar a ser un buen profesional.

TAREAS TÉCNICAS

- La realización de un estudio de los diferentes métodos de control de activos fijos tangibles.
- La realización de un estudio de los Códigos de Barras y su utilización en las diferentes esferas de la economía.
- La realización de un estudio del desarrollo de software sobre C# de la plataforma .NET.
- La realización de un estudio sobre el desarrollo de bases de datos en SQL Server 2005.
- La evaluación de la efectividad y viabilidad del sistema de control de activos fijos tangibles diseñado.
- El desarrollo de un software y una base de datos para el control de activos fijos tangibles.
- El desarrollo de un manual de usuario para el software.

Firma del Autor

Firma del Tutor

RESUMEN

El presente trabajo esta dedicado a la realización de un sistema para automatizar el proceso de control de activos fijos tangibles en la facultad de Ingeniería Eléctrica de la Universidad Central “Marta Abreu” de Las Villas pudiéndose extender a cualquier empresa o entidad laboral. Para esto se confeccionará un software en C# 2.0 el cual consta con una interfaz amigable y de fácil manejo para el usuario. Una base de datos en SQL Server 2005 donde se almacenaran todos los datos referentes a cada uno de los activos fijos y que pueda ser consultada por el usuario mediante el software.

También se utilizarán códigos de barra, para identificar a los activos fijos, en los cuales estará codificado el número de inventario de cada uno de los activos. El software permitirá al usuario intercambiar información con la base de datos y realizar una serie de operaciones necesarias para el control y chequeo de activos fijos tangibles.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
TAREA TÉCNICA	iv
RESUMEN	v
INTRODUCCIÓN	1
Organización del informe.	3
CAPÍTULO 1. INTRODUCCION AL PROYECTO.....	4
1.1 Descripción del proyecto.....	4
1.2 Control de activos fijos en la UCLV.....	4
1.3 Inconvenientes del proceso manual.....	5
1.4 Solución propuesta.....	5
1.5 Beneficios que ofrece la solución.....	6
1.6 Elementos necesarios para la realización de la solución.....	7
1.7 ¿Qué son los códigos de barras?.....	7
1.8 Herramientas utilizadas.....	8
1.8.1 Gestor de Base de Datos SQL Server 2005.....	8
1.8.2 Beneficios ofrecidos por SQL Server 2005.....	9

1.8.3 Lenguaje de programación C# 2.0.....	9
1.8.4 Características de C# 2.0.....	10
1.8.5 Ventajas que brinda.....	11
1.9 Descripción del diseño arquitectónico del sistema de información.....	12
CAPÍTULO 2. DESCRIPCIÓN DEL SISTEMA AUTOMATIZADO PARA EL CONTROL DE ACTIVOS FIJOS TANGIBLES.....	14
2.1 Diseño de la base de datos.....	14
2.1.1 Diseño conceptual.....	18
2.1.2 Diseño lógico.....	19
2.1.3 Diseño físico.....	20
2.2 Diseño del software.....	22
2.2.1 La capa de presentación o aplicación.....	23
2.2.2 La capa de negocio.....	26
2.2.3 La capa de acceso a datos.....	28
2.3 Código de barra seleccionado.....	28
2.4 Control automatizado de los activos fijos tangibles.....	29
CAPÍTULO 3. MANUAL DE USUARIO Y PRUEBAS AL SISTEMA.....	300
3.1 Manual de usuario y pruebas al sistema.....	30
CONCLUSIONES Y RECOMENDACIONES.....	37
Conclusiones.....	37
Recomendaciones.....	38
REFERENCIAS BIBLIOGRÁFICAS.....	39
ANEXOS.....	41

Anexo I	El Código de	
Barras.....		411
Anexo II	Modelo para el control de activos fijos tangibles.....	46

INTRODUCCIÓN

En estudios exploratorios se ha podido constatar, que en la mayoría de los centros laborales de nuestro país no se utilizan sistemas automatizados en los procesos de control de activos fijos o activos de planta como también se les conoce. Esto trae como consecuencia mayor costo en la utilización de los recursos humanos y materiales, además, el empleo de mayor cantidad de tiempo en esta actividad.

Los activos de planta son aquellos de larga vida que son tangibles, por ejemplo, los terrenos, los edificios, los equipos eléctricos y los muebles. Se controlan en forma muy parecida a como se controlan los inventarios de alto precio con registros auxiliares. Cada uno de estos activos está representado por una tarjeta que lo describe y relaciona su ubicación y el empleado responsable del mismo.

Este trabajo tiene como propósito contribuir al perfeccionamiento del proceso de control de activos fijos tangibles en la facultad de Ingeniería Eléctrica, para ello se utilizarán códigos de barra para identificar a cada uno de los medios existentes, o sea, su número de inventario estará codificado y puesto en una etiqueta sobre el activo fijo correspondiente, además se contará con una base de datos para guardar todo lo referente a cada uno de los medios y un software para controlar e interactuar con dicha base de datos. Así esta actividad será más confiable, práctica y económica.

Objetivo:

Elaborar un software, con interfaz cómoda y amigable, fácilmente extensible, con código ejecutable para MS Windows, utilizando el lenguaje C# 2.0 y el SGBD MS SQL Server

2005, para la automatización del proceso de control de activos fijos tangibles en la facultad de Ingeniería Eléctrica.

Objetivos específicos:

- Realizar un estudio del estado del arte de los diferentes métodos de control de activos fijos tangibles.
- Realizar un estudio del estado del arte del desarrollo de software en C# 2.0 de la plataforma .NET.
- Realizar un estudio del estado del arte de la metodología para el diseño y elaboración de bases de datos en SQL Server 2005.
- Realizar un software para la automatización del proceso de control de activos fijos tangibles.

Hipótesis de la investigación.

Si se elabora un software para la automatización del proceso de control de activos fijos tangibles de una institución o centro laboral, se logrará que esta actividad sea más confiable, práctica y económica.

Organización del informe.

El informe de la investigación se estructurará en introducción, capitulario, conclusiones, referencias bibliográficas y anexos.

En la introducción se dejará definida la importancia, actualidad y necesidad del tema que se aborda.

Desarrollo.

CAPITULO I: Se dedicará a la presentación de todo el basamento teórico, arquitecturas de los sistemas de información orientados a objeto, los códigos de barras, el SGBD SQL Server 2005 y el lenguaje de programación C#.

CAPITULO II: Se utilizará para explicar el desarrollo de la investigación. Descripción de la arquitectura del sistema y el diseño de la base de datos.

CAPITULO III: Se dedicará a expresar los resultados prácticos del desarrollo. Se brinda un manual de usuario.

CAPÍTULO 1. INTRODUCCION AL PROYECTO

1.1 Descripción del proyecto

Toda empresa o entidad laboral necesita, de una manera u otra, tener control de todos sus medios físicos para evitar pérdidas, contabilizarlos y mantener actualizado el inventario de estos. De este proceso se encarga el control de activos fijos tangibles, que hace algún tiempo se realizaba manualmente, pero con el auge de informatización que existe en el mundo ya se está automatizando, aunque no todas las empresas o entidades laborales lo han adoptado y solo unas pocas marcan la vanguardia en esta área. Investigaciones realizadas arrojaron que en nuestro país el control de activos fijos tangibles se realiza de forma manual en casi todas las entidades o empresas existentes. Solo unas pocas, como algunas dependencias del ministerio de salud, utilizan algún tipo de sistema para la automatización de este proceso.

1.2 Control de activos fijos en la UCLV.

Al realizar un estudio en varias áreas de la Universidad Central “Marta Abreu” de Las Villas, con vistas a obtener información sobre el proceso, se detectó que el proceso de control de activos fijos se realiza de forma manual en todas dependencias. Entrevistando a los encargados de ese proceso se elaboró una relación de todas las acciones que se llevan a cabo en esta tarea. A continuación se muestran todas las operaciones que se realizan en este proceso manual:

- La persona encargada va hasta el edificio administrativo y recoge el modelo impreso con la relación de activos fijos tangibles correspondientes a su área.
- El modelo recogido es pasado a tablas de Microsoft Excel para su posterior utilización y consulta.
- El modelo en Excel es pasado a un documento Word ya que las tablas para el chequeo y control de los activos fijos tangibles están en este formato.
- Todos los meses se recibe por correo electrónico la relación de los medios para efectuar el control del 10%, que es chequeado manualmente por el encargado del control en cada área.
- En el momento del chequeo, el encargado debe ir a cada uno de los locales controlando los números de inventario de cada uno de los activos existentes.

1.3 Inconvenientes del proceso manual

Analizando la realización del proceso se pueden encontrar una serie de inconvenientes y limitaciones que entorpecen la realización de este proceso, así como su fiabilidad. Las principales limitantes son:

- Enviar una persona a recoger los modelos nuevos al edificio administrativo.
- Digitalizar los modelos y convertirlos a varios formatos.
- Chequear de forma manual la lista de medios existentes, impresa en papel, contra los que hay en el local.
- Cada vez que se mueve o asigna un nuevo medio hay que imprimir todos los modelos nuevamente.

1.4 Solución propuesta.

Para erradicar todas estas deficiencias se decidió elaborar un software para la automatización del control de activos fijos tangibles, lo que implicaría rapidez, eficiencia y abaratamiento del proceso. Logrando búsquedas más rápidas y eficientes de los datos de cada activo fijo (descripción, área a la que pertenece, fecha de alta, etc.), chequeo más

preciso de los medios existentes contra los asignados y actualizar las bases de datos existentes, sin tener que imprimir de nuevo todos los modelos con los medios actualizados. Además el proceso quedará libre de errores humanos y aumentara considerablemente su fiabilidad.

Para lograr esto, el sistema deberá mostrar la relación de todos los activos fijos existentes, permitir al usuario realizar búsquedas específicas según los parámetros ofrecidos, mostrando solo los activos fijos que coincidan con la selección del usuario. Actualizar los campos del medio que el usuario desee, agregar y borrar un medio de la base de datos a petición del usuario. Para alcanzar una automatización total de este proceso es necesario que cada activo fijo tenga un código de barra con su número de inventario, el cual va a ser leído por un scanner o lector de códigos de barra.

Por eso, el presente proyecto está dedicado a la realización de un sistema para automatizar el control de activos fijos tangibles en la facultad de Ingeniería Eléctrica de la UCLV, empleando Código de Barras para identificar a los mismos.

1.5 Beneficios que ofrece la solución.

Con la solución propuesta se ofrecen una serie de beneficios tanto a los usuarios como a la facultad. Dentro de estos se puede destacar el ahorro de tiempo en el procesamiento de los datos y control de los medios, ahorro en papel y tóner de impresión ya que solo se imprimirían los modelos necesarios para archivar ya que todos los cambios y movimientos de activos fijos se realizan en formato digital. Los datos se almacenan en un servidor central donde los jefes de cada área los pueden consultar desde su puesto de trabajo. Aumentaría notablemente la eficiencia y eficacia del proceso gracias a que se eliminan los controles visuales de los códigos de los activos, porque solamente sería leerlos con un scanner de códigos de barra. De esta forma se eliminan también los errores humanos en el proceso de control.

1.6 Elementos necesarios para la realización de la solución.

Para este proyecto es necesario tener una base de datos almacenada en un servidor, un software que le permita al usuario conectarse e interactuar con la base de datos para agregar, borrar, modificar, chequear y buscar medios en esta. Además es necesario identificar cada uno de los activos fijos con un código de barras en el cual va codificado el número de inventario de este y tener un scanner de códigos de barra para leerlos.

1.7 ¿Qué son los códigos de barra? ^{[1], [2], [3], [4]}.

Los Códigos de Barras son una forma fácil, rápida y precisa de codificar información, estos aparecen impresos en las diferentes etiquetas que se utilizan en alimentos, paquetes de mercancías, lo mismo en los supermercados, tiendas por departamentos, farmacias, brazaletes de pacientes, o sea, los Códigos de Barras se han insertado en nuestras vidas sin que muchos de nosotros sepamos lo que significan o representan. Lo que sí está claramente definido es, que aunque todos parezcan iguales, hay gran variedad de estándares que son utilizados y aplicados por cada instancia, en dependencia de sus requerimientos o necesidades particulares. Para mayor descripción ver Anexo 1.

Tipos de códigos de barra.

Existen gran variedad de códigos de barra y cada uno tiene su aplicación específica para lo que fue diseñado, dentro de esta gran variedad se destacan los siguientes, por ser los más vistos en la vida común.

- **Código 39:** es utilizado en el inventariado de objetos y mercancía.
- **Código 128:** una variante del código 39 pero permite codificar mayor cantidad de información.
- **Código Codabar:** de gran aplicación en los bancos de sangre.
- **Universal Product Code (U.P.C.):** utilizada en el comercio minorista en EEUU, solo permite codificar números.

1.8 Herramientas utilizadas.

En el desarrollo de este trabajo fue necesario elegir entre varias herramientas de software, las más factibles y utilizadas en el mundo, y que mejor se ajustaran a nuestras necesidades, esto llevó a seleccionar C# de la plataforma .NET para desarrollo de la aplicación y SQL Server 2005 para desarrollar la base de datos.

1.8.1 Gestor de Bases de Datos SQL Server 2005 ^[5].

SQL Server 2005 fue diseñado para ayudar a sus usuarios a enfrentar grandes desafíos, tales como, la necesidad de tomar decisiones más rápidas y más orientadas a datos, aumentar la productividad y flexibilidad del personal de desarrollo y reducir los presupuestos generales de informática para satisfacer las exigencias cada vez mayores de las empresas en desarrollo. Esta solución de administración y análisis de datos de próxima generación ofrece mayor seguridad, escalabilidad y disponibilidad a las aplicaciones de datos empresariales, a la vez que las hace más fáciles de crear, desplegar y administrar.

Con su gran cantidad de ventajas SQL Server 2005 integra administración y análisis de datos, lo que ayuda a las organizaciones y entidades de cualquier magnitud a:

- Maximizar la productividad mediante la reducción de la complejidad y el soporte de aplicaciones de bases de datos.
- Crear, desplegar y administrar aplicaciones empresariales más seguras, escalables y confiables.
- Compartir datos en distintas plataformas, aplicaciones y dispositivos facilitando la conexión de sistemas internos y externos.
- Controlar los costes sin sacrificar en rendimiento, disponibilidad, escalabilidad y seguridad.

1.8.2 Beneficios ofrecidos por SQL Server 2005.

La plataforma de datos SQL Server 2005 ofrece los siguientes beneficios a las organizaciones de todas las magnitudes:

- **Aprovechamiento de activos de datos:** Permite que los clientes obtengan más valor de sus datos al incluir funcionalidades tales como informe, análisis y minería de datos. Se puede aprovechar esta potencia y flexibilidad para entregar datos a cada rincón de su organización o empresa con mucho menos costo que el de otros sistemas ^[6].
- **Aumento de la productividad:** Mediante la integración con herramientas conocidas como Microsoft Office System, SQL Server 2005 brinda a los usuarios en toda su organización información empresarial crítica y oportuna adaptada a sus necesidades específicas. El objetivo ayudar a los usuarios en todos los niveles de la organización a tomar mejores decisiones empresariales según uno de sus activos más valiosos: los datos.
- **Menor coste total de propiedad:** El enfoque y la atención integrados sobre la facilidad de uso y despliegue en SQL Server 2005 ofrece los costes directos de implementación y mantenimiento más bajos de la industria, para obtener un rápido rendimiento de su inversión en la base de datos.

1.8.3 Lenguaje de programación C# 2.0.

Desde la aparición de las computadoras, son incontables los lenguajes de programación que han surgido y aunque unos son mejores que otros, todos tienen sus ventajas y sus desventajas. La desventaja fundamental es que entre ellos existen maneras muy diferentes de estructurar el código, además de que cada programa maneja sus propias librerías y sintaxis, además de que las funciones en algunos casos se tienen que crear desde cero, lo que conlleva mas tiempo a la hora de programar ^[7].

En su afán por solucionar esto la empresa de Microsoft creo un lenguaje que podría ser la respuesta a todo este tipo de inconvenientes, un lenguaje llamado C# que es uno de los

lenguajes base de su plataforma .NET. Lo que pretende este lenguaje es ser muy versátil en su uso y eficiente en su aplicación conformando un lenguaje que reúne las mejores características de los lenguajes más utilizados y agrupándolos en un solo lenguaje mejorado.

1.8.4 Características de C# 2.0.

- **Facilidad de uso:** Presenta un ambiente de trabajo muy cómodo y amigable, clásico de las aplicaciones de Windows. Su forma de programar será fácil de usar para quien esté familiarizado con C++, ya que su estructura básica es muy similar.
- **Programación orientada a objetos:** Esta forma de programación ahorra mucho código, dado que hay partes del código que son reutilizables y no hay que volverlas a escribir, por esto C# es un lenguaje orientado a objetos, con las siguientes propiedades: encapsulación, polimorfismo y herencia simple.
- **Administración de memoria:** C# tiene la característica de inicializar los datos o variables declaradas en el programa de forma automática y liberar la memoria cuando el mismo programa lo cree conveniente [8]. Es decir tiene constructores y destructores, y estos actúan automáticamente a menos que se manipulen desde el código.
- **Seguridad en el manejo de datos:** Los tipos de datos que se manejan se comprueban para asegurarse que se correspondan a los validados para las funciones que han sido creadas; así también se vigila que no se produzcan errores en operaciones matemáticas, además de que se impide el uso de variables que no han sido inicializadas. Todo esto permite que no se produzcan errores en el momento de la ejecución.
- **Sistema de tipos unificado:** Todos los tipos de datos que se definan siempre se derivarán, incluso de forma implícita, de una clase base común llamada *System.Object*, por lo que dispondrán de todos los miembros definidos en ésta clase.

- La ventaja de que todos los tipos se deriven de una clase común es que facilita el diseño de colecciones genéricas que puedan almacenar objetos de cualquier tipo.
- **Uso de operadores:** Este lenguaje permite de forma automática la manera en que pueden trabajar los operadores, ya sea de tipo lógico o aritmético. Es decir dependiendo del contexto de donde se encuentre el operador, el programa detecta que tipo de uso debe tener el mismo.
- **Compatible:** Al mantenerse en C# una sintaxis muy similar a C, C++ o Java se permite incluir directamente fragmentos de código escrito en estos lenguajes, además ofrece la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32.

1.8.5 Ventajas que brinda ^{[8],[9]}.

Las ventajas que ofrece C# frente a otros lenguajes de programación son:

- **Declaraciones en el espacio de nombres:** Es posible definir una o más clases dentro de un mismo espacio de nombres.
- **Tipos de datos:** Posee un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- **Atributos:** Todos los miembros de una clase tienen un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.
- **Pase de parámetros:** Los métodos pueden ser declarados para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada *ref* para el pase de parámetros por referencia.
- **Métodos virtuales y redefiniciones:** Los métodos pueden ser redefinidos en una clase base siempre que sean declarados como virtuales con anterioridad. El método redefinido en la subclase debe ser declarado con la palabra *override*.

- **Propiedades:** Un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, se 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.

1.9 Descripción del diseño arquitectónico del sistema de información ^[10].

En la actualidad el diseño de sistemas de información está basado en los diseños por capas, en donde el objetivo primordial es separar la capa de datos, la capa de presentación al usuario y la capa de negocio.

La ventaja principal que ofrece este estilo, es que el desarrollo se puede llevar a cabo en varias capas y en caso de algún cambio sólo se ataca la capa requerida sin tener que revisar entre código mezclado. El diseño más utilizado actualmente es el diseño en tres capas las cuales son:

- **Capa de presentación o aplicación:** es la que se le muestra al usuario y con la que este interactúa directamente. Esta capa le comunica la información al usuario y captura sus acciones realizando un filtrado previo para comprobar que no hay errores de formato. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio:** aquí residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él ^[11].
- **Capa de datos:** aquí se almacenan los datos. Está formada por uno o más gestor de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

El sistema de control de activos fijos tangibles es un sistema que permite al usuario acceder a datos almacenados en una base de datos de una manera cómoda y rápida, ya que presenta una interfaz gráfica de usuario, una interfaz de acceso a datos y una base de datos. En la figura 1 se muestra un esquema de este tipo de sistemas, basado en la arquitectura de tres capas.



Fig. 1.1 Arquitectura de tres capas de un sistema de información orientado a objetos.

CAPÍTULO 2. DESCRIPCIÓN DEL SISTEMA AUTOMATIZADO PARA EL CONTROL DE ACTIVOS FIJOS TANGIBLES.

En este capítulo se explica de manera detallada el diseño de la base de datos, los procedimientos almacenados que existen en ella y las relaciones que la conforman. Además se explica el diseño por capas que componen la aplicación, describiendo clases, métodos y propiedades.

2.1 Diseño de la base de datos

Cuando se desarrolla una aplicación es necesario tener bien claras las descripciones de la solución lógica para saber como satisfacen esta los requerimientos y las restricciones del problema. La fase de diseño deja bien claro una cosa, cómo el sistema cumple con los requerimientos planteados.

2.1.1 Diseño conceptual

Análisis de los requisitos: se desea establecer un modelo conceptual para la gestión del control de activos fijos tangibles de un centro laboral. De cada medio es necesario conocer su número de inventario, el nombre y el código a de la entidad a la que pertenece, el nombre y código del área, la descripción, la fecha de alta, la fecha de baja, el valor en moneda nacional y en CUC y la fecha de emisión del documento. Todo esto para cuando el

usuario realice una búsqueda o chequeo de los medios en existencia le brinde el estado actual del medio en cuestión.

2.1.2 Diseño lógico

El diseño lógico de una base de datos consta de dos etapas: el diseño lógico estándar y el diseño lógico específico. En el diseño lógico estándar, se toma el esquema conceptual resultante de la fase de diseño conceptual y teniendo en cuenta los requisitos del proceso, se construye un esquema lógico estándar (ELE), que se apoya en un modelo lógico estándar (MLE), que será el mismo modelo de datos soportado por el SGBD a utilizar pero sin restricciones de ningún producto comercial en concreto. El diseño lógico específico esta muy ligado a la fase de diseño físico.

A continuación se muestra el esquema lógico resultante de la base de datos.

	Column Name	Data Type	Allow Nulls
	NoDeInventario	bigint	<input type="checkbox"/>
	NombreDeLaEntidad	nvarchar(50)	<input checked="" type="checkbox"/>
	CodigoDeLaEntidad	nvarchar(50)	<input checked="" type="checkbox"/>
	NombreyCodigoDelArea	nvarchar(50)	<input checked="" type="checkbox"/>
	FechaDeEmision	nvarchar(50)	<input checked="" type="checkbox"/>
	Descripcion	nvarchar(50)	<input checked="" type="checkbox"/>
	FechaAlta	nvarchar(50)	<input checked="" type="checkbox"/>
	ValorCUC	float	<input checked="" type="checkbox"/>
	ValorMN	float	<input checked="" type="checkbox"/>
	FechaBaja	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Fig. 2.1 Esquema lógico de la base de datos Control.

La relación Control contiene los datos relativos a cada una de los activos fijos y define como llave primaria NoDeInventario por lo que este parámetro nunca podrá pasarse nulo desde el software, los demás parámetros si aceptan *null* por defecto, dando así la opción al

usuario de no tener que llenarlos en el instante que se agrega un nuevo activo fijo a la base de datos.

Descripción de los campos de la tabla:

- *NoInventario*: es donde se almacena el número de inventario correspondiente a cada activo fijo, este dato es del tipo *bigint* (número entero).
- *CodigoDeLaEntidad*: se guarda el código de la entidad a la que pertenece el activo fijo, este dato tiene que ser del tipo *nvarchar(50)* (cadena de hasta 50 caracteres).
- *NombreyCodigoDelArea*: se guarda el nombre y el código del área donde se encuentra el activo fijo, también es del tipo *nvarchar(50)*.
- *FechaDeEmision*: es la fecha en la que se emite el documento, es *varchar(50)*. Debería ser del tipo *datetime*, que es la variable que se asigna a las fechas, pero como no hay un formato estándar para escribir la fecha de emisión, no se puede limitar al usuario a entrarlo en formato de fecha en *SQL*.
- *Descripcion*: es la descripción que presenta el activo fijo. *varchar(50)*.
- *FechaAlta*: la fecha en que se da de alta al activo fijo. Ocurre lo mismo que en *FechaDeEmision* con el tipo de dato.
- *ValorMN* y *ValorCUC*: son los valores que tiene el activo fijo en moneda nacional y moneda libremente convertible. Son del tipo *float* que permite representar un dato con coma flotante.
- *FechaBaja*: la fecha en la que se le da de baja al activo fijo tangible, ocurre lo mismo que en los dos caso anteriores con el tipo de dato.

En las figuras siguientes se muestran los diagramas de bloques de las acciones permitidas para el usuario en la base de datos. Estas acciones dan la posibilidad de que el usuario obtenga, modifique y agregue los datos necesarios en la tabla.

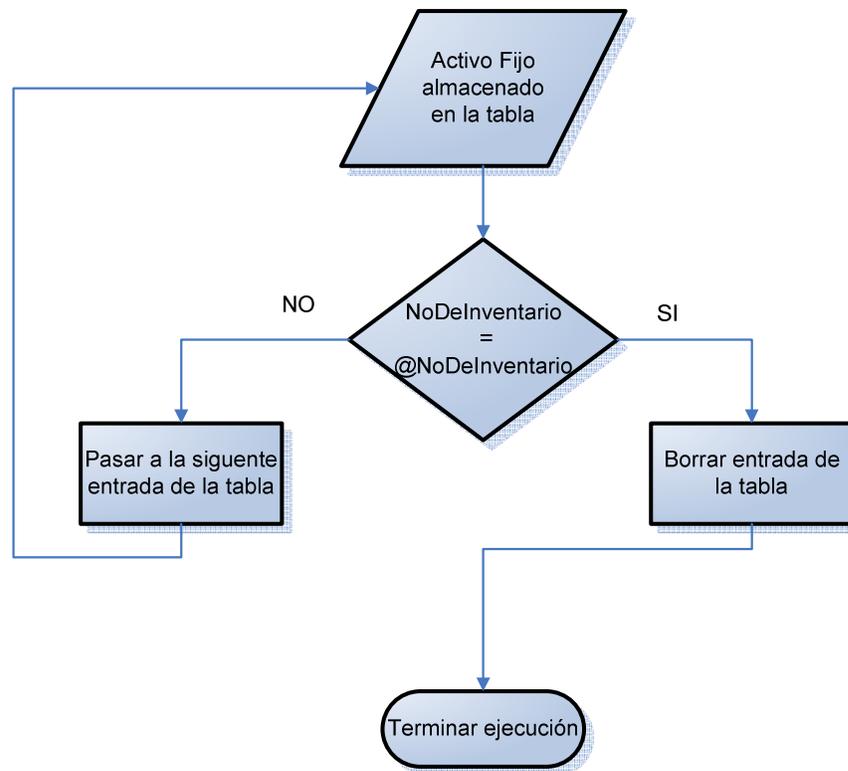


Fig. 2.2 Diagrama del procedimiento almacenado *BorrarATF*.

En este diagrama corresponde al proceso de borrado de un activo fijo de la base de datos. Primero se toma el número de inventario (*NoInventario*) del primer activo fijo almacenando en la tabla y se compara con el número de inventario de del activo que se quiere borrar, si son iguales se borra esa entrada de la tabla y se termina la ejecución. En caso de ser diferentes se pasa al siguiente activo en la tabla y se ejecuta el mismo proceso.

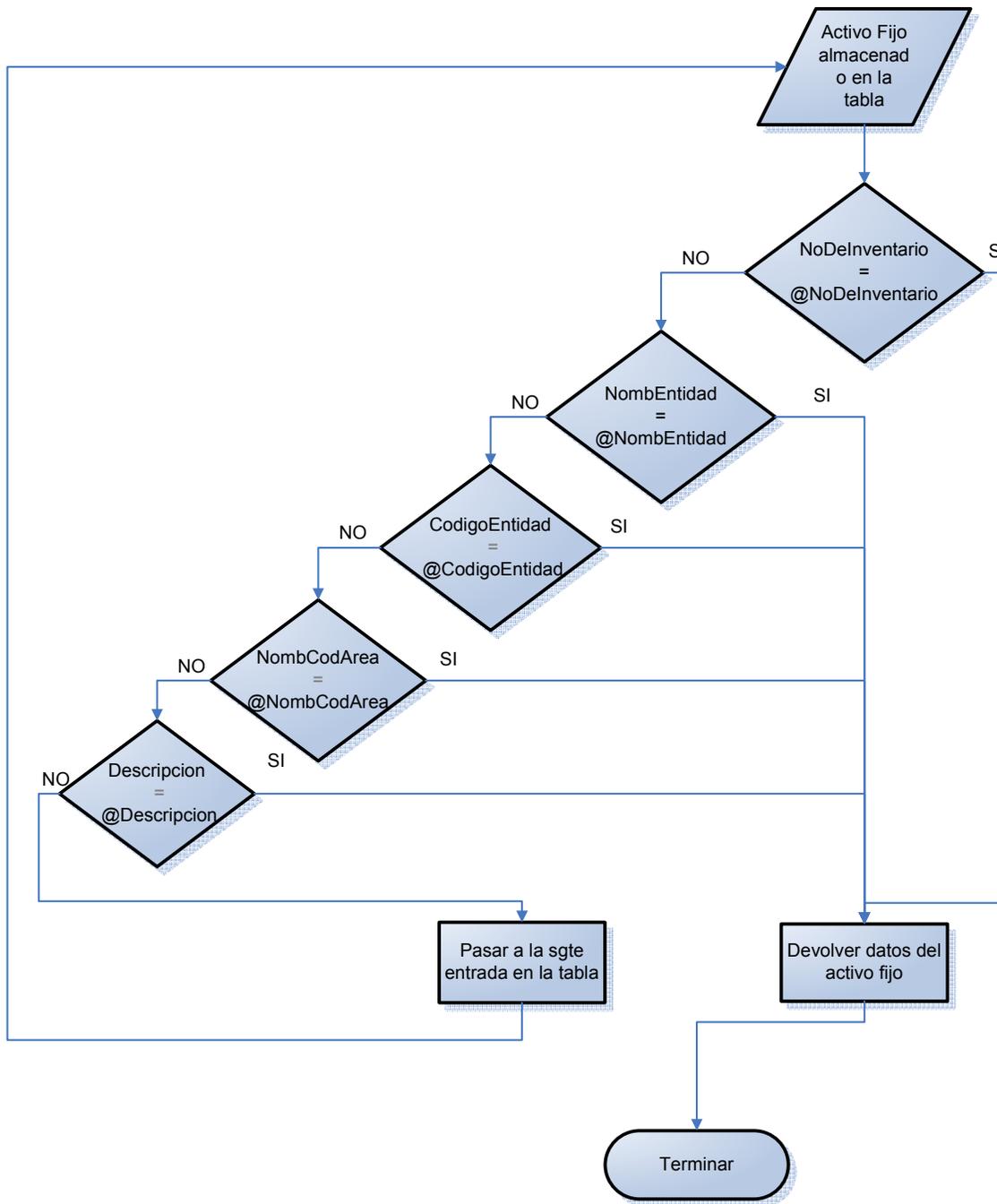


Fig. 2.3 Diagrama del procedimiento *BuscarATF*.

En este diagrama se comienza por obtener de la base de datos los parámetros *NoInventario*, *NombreEntidad*, *CodigoEntidad*, *NombreCodigoArea* y *Descripcion* de la primera entrada

de la tabla y comenzar por comparar *NoInventario* con *@NoInventario* (parámetro de búsqueda), si son iguales se devuelven los datos de todos los campos de esa entrada, si no, se pasa a comparar *NombreEntidad* con *@NombreEntidad* (parámetro de búsqueda) y así sucesivamente. En caso de no encontrarse coincidencia en los campos, se pasa a la siguiente entrada en la tabla y se repite la operación.

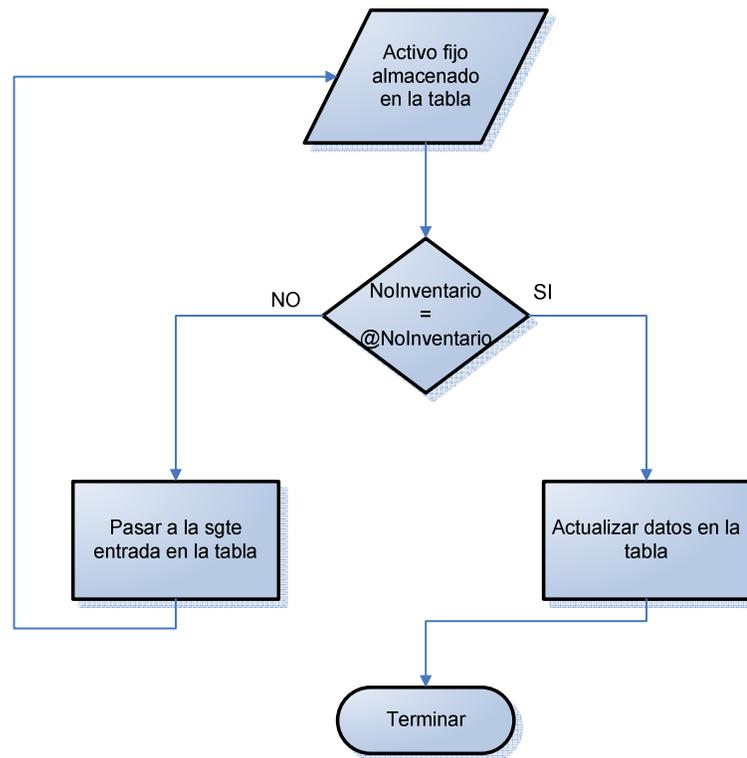


Fig. 2.4 Diagrama del procedimiento almacenado *ModificarMedio*.

En este caso se compara el campo *NoInventario* de la primera entrada de la tabla con *@NoInventario* (número inventario del activo fijo modificado), si coinciden se actualizan todos los campos de esta entrada en la tabla, si no, se pasa a la siguiente entrada y se repite la operación.

Descripción de los procedimientos almacenados.

Para realizar todas las operaciones anteriormente explicadas fue necesario crear en la base de datos una serie de procedimientos almacenados que se encargaran de interactuar y manejar los datos. Estos procedimientos se explican a continuación:

- *LeerMedios*: se encarga de proporcionar al software todos los datos de cada una de las entradas de la tabla, como número de inventario, nombre de la entidad, nombre del área, número y código del área, descripción, fecha de emisión, fecha de alta, fecha de baja, valor en moneda nacional y valor en moneda libremente convertible.
- *BorrarATF*: en caso de que se desee borrar un medio de la tabla, se le pasa el número de inventario del medio seleccionado y el procedimiento almacenado elimina esa entrada de la tabla.
- *BuscarATF*: se encarga de buscar uno o varios medios dentro de la tabla y devuelve los datos de las entradas encontradas que coincidan con los parámetros establecidos en la búsqueda, como número de inventario, nombre de la entidad, código de la entidad, nombre y código del área o descripción.
- *AgregarATF*: se encarga de añadir una nueva fila a la tabla con los valores recibidos, el valor número de inventario no puede ser *null* puesto que es la llave primaria de la tabla y no permite agregar una nueva entrada sin este valor.
- *ModificarMedio*: este procedimiento es el encargado actualizar los datos de una entrada en la tabla cuyo número de inventario se corresponda con el que se recibe.

2.1.3 Diseño físico.

Es la expresión material del diseño lógico, incluye las cantidades y cualidades del hardware, software, y sistemas de aplicación. Además de su distribución geográfica local o amplia.

Requerimientos:

Los requerimientos están condicionados por las necesidades del SGBD y el Framework que se van a utilizar, en este caso, Microsoft SQL Server 2005 y Framework 2.0.

Tabla 1. Requerimientos.

Procesador	<ul style="list-style-type: none"> ▪ Intel o procesadores Pentium III compatibles a 500 MHz o más.
Requerimientos de memoria RAM para los distintos SGBD	<ul style="list-style-type: none"> ▪ SQL Server 2005 Enterprise Edition: mínimo 256 MB, recomendado 512 MB. ▪ SQL Server 2005 Standard Edition: mínimo 256 MB. ▪ SQL Server 2005 Personal Edition: mínimo 256 MB.
Requerimientos de Disco Duro	<ul style="list-style-type: none"> ▪ 350 MB de espacio disponible en el disco duro para la instalación completa del SGBD SQL Server 2005. ▪ 390 MB para bases de datos de muestra.
Video	<ul style="list-style-type: none"> ▪ Monitor con resolución VGA. Las herramientas graficas de SQL Server 2005 requieren un monitor con resolución de 800x600 o superior.
Hardware	<ul style="list-style-type: none"> ▪ Torre de CD-ROM drive, y un <i>Mouse</i> Microsoft u otro dispositivo compatible.
Otros	<ul style="list-style-type: none"> ▪ SQL Server 2005 requiere Internet Explorer 5.0 o mayor y se soporta en los sistemas operativos: ▪ Windows 95, Windows 98, Windows Me, Windows NT 4.0, Windows 2000 y Windows XP poseen software de red incorporado. Se requiere software de red adicional si utiliza Banyan VINES o AppleTalk ADSP. El protocolo NWLink

	de red basada en Windows provee soporte al cliente Novell Netware IPX/SPX.
--	--

2.2 Diseño del software.

El software fue diseñado sobre la arquitectura de tres capas por lo que en este capítulo se explica cada una de ellas y su relación entre si. Está compuesto por una solución que se llama *ControlMedios* la cual está formada por tres proyectos, *Control*, *Reglas del Negocio* y *Acceso a Datos*. Las bibliotecas de clase *ReglasNegocio.dll* y *AccesoDatos.dll* son generadas por los proyectos *Reglas del Negocio* y *Acceso a Datos*, el ejecutable *Control.exe* es generado por el proyecto *Presentación* y este utiliza las dos *DLLs* anteriores como se muestra en la figura.

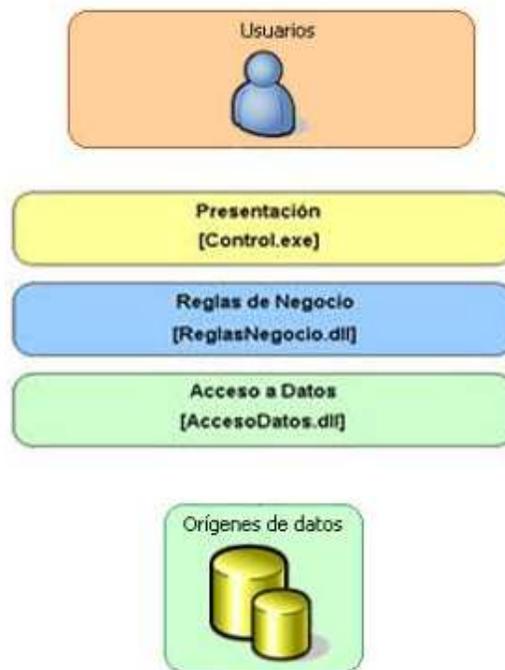


Fig. 2.5 Estructura del software basado en la arquitectura por capas.

2.2.1 La capa de presentación o aplicación.

En la capa de presentación es donde se encuentra la interfaz de usuario para facilitar la relación de este con el software y las bases de datos, mediante esta interfaz el usuario puede realizar una serie de acciones sobre la base de datos, pero no directamente, ya que esta capa solo interactúa con la capa de negocios y esta es la encargada de tramitar con la capa de datos las acciones realizadas por el usuario sobre la base de datos.

Este proyecto está conformado por *PresentacionForm* y *OrigenCodigosForm*, los cuales son dos formularios para que el usuario pueda interactuar con el programa. *PresentacionForm* es la forma que se carga por defecto y donde aparecen los datos cuando se ejecuta por primera vez el programa y con los que el usuario trabajaría. La forma *OrigenCodigosForm* solo es visible cuando el usuario intenta chequear los medios existentes contra los obtenidos en los locales dándole la posibilidad a este de especificar por qué vía va a ingresar estos datos. Estas pueden ser: leer directamente desde el scanner, desde un archivo de texto en una dirección dada o entrados manualmente en el momento de la ejecución.

Mediante la interfaz de usuario será posible realizar una serie de acciones, tales como, buscar, agregar, borrar, chequear y refrescar los datos en el programa. Esta capa está formada por cinco *cajas de texto* y un *datagrid*, que es la forma de representar tablas de manera visual en C#. En el *datagrid* es donde se muestran los datos de cada uno de los activos fijos contenidos en la base de datos, de manera tal que el usuario puede interactuar de manera fácil con ellos. Las *cajas de texto* son para ingresar parámetros para realizar búsquedas dentro de la base de datos, solo mostrando los que coincidan con los datos ingresados. Sobre el *datagrid* también se pueden realizar una serie de operaciones que quedarían registradas en la base de datos, como pueden ser borrar, agregar, buscar o modificar un medio. Los métodos encargados de estas operaciones se muestran continuación:

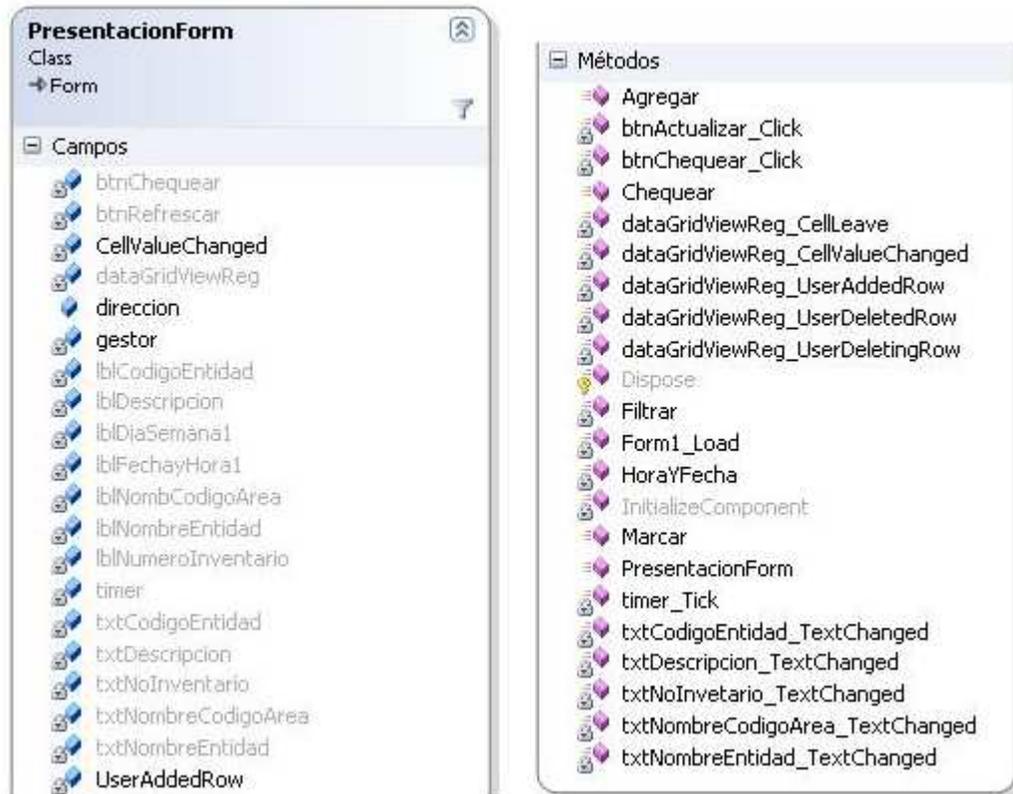


Fig. 2.6 Diagrama de clase de la capa de presentación o aplicación.

Cada uno de estos métodos se comentan a continuación:

- *Filtrar*: este método es llamado dentro de los eventos *txtNoInventario_TextChanged*, *txtNombreEntidad_TextChanged*, *txtCodigoEntidad_TextChanged*, *txtNombreCodigoArea_TextChanged*, *txtDescripcion_TextChanged* que son generados cuando el texto de alguno de los cuadros de texto cambia. Los valores son enviados capa de negocio (será explicada más adelante), para que se ejecute la búsqueda en la base de datos de los medios correspondientes.
- *dataGridViewReg_UserDeletingRow*: detecta cuando el usuario intenta eliminar una fila, mostrando un cuadro de diálogo para que este confirme su acción, así se evita borrar un medio por equivocación.

- *dataGridViewReg_UserDeletedRow*: este evento ocurre cuando se eliminó la fila y es el encargado de refrescar los datos dentro del *dataGridView* para que muestre las filas restantes.
- *dataGridViewReg_UserAddedRow*: este evento ocurre cuando se agregan datos en una nueva fila en el *datagrid*, asignándole a la variable *UserAddRow* el valor *True*.
- *dataGridViewReg_CellValueChanged*: ocurre cuando el valor de una celda es modificado, en este evento se chequea si la variable *UserAddRow* es *True* (*verdadero*), en caso de serlo se está indicando que fue una celda dentro de una nueva fila, en este caso se envían los nuevos datos a la capa de Reglas del Negocio para que sea insertada esta fila dentro de la base de datos. Si *UserAddRow* es *False* (*falso*) solo se pasarían los datos modificados para que sean actualizados dentro de la tabla en la base de datos.
- *Chequear*: se encarga de comparar los códigos correspondientes a los números de inventario de los medios de la base de datos contra los leídos en los locales que fueron ingresados por el usuario, si se detecta que falta algún medio se llama al método *Marcar* que marcaría la fila en color rojo.

Dentro de la clase *Control* se han definido una gran cantidad de métodos los cuales son los encargados de realizar las acciones que el usuario necesite para el control de activos fijos. También se utilizan métodos que ya están implementados en clases del sistema, por ejemplo los encargados de mostrar la hora y la fecha.

El evento *timer_tick* ocurre cada un segundo y dentro de el se llama el método *HoraYFecha* que utiliza un objeto *DateTime* del sistema para mostrar en los *labels lblDiaSemana1* y *lblFechayHora1* la fecha correspondiente obtenida del sistema.

2.2.2 La capa de negocio

En esta capa se definieron cinco métodos los cuales son los encargados de recibir los datos generados por las acciones de los usuarios y entregárselos a la capa de datos para que esta los ejecute sobre la base de datos.

A continuación se muestra el diagrama de clase de esta capa:

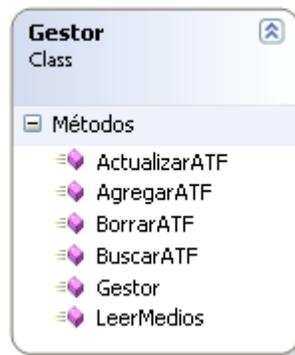


Fig. 2.7 Diagrama de clase de *Gestor.cs*.

- *LeerMedios*: es la encargada de buscar y entregarle a la capa de aplicación la relación de todos los activos fijos existentes en la base de datos cuando se ejecuta por primera vez el programa o a petición del usuario cuando se pide actualizar la tabla.
- *BuscarATF*: se encarga de pedirle a la base de datos los activos fijos que concuerden con los datos introducidos por el usuario para la realización de la búsqueda, y devuelve a la capa de aplicación los resultados obtenidos.
- *AgregarATF*: esta función le envía a la base de datos el número de inventario del nuevo activo fijo ingresado para que pueda ser agregado.
- *ActualizarATF*: pasa a la base de datos los valores modificados en cada uno de los campos de un activo fijo para que esta se actualice y devuelve a la capa de aplicación los nuevos valores.

- *BorrarATF*: envía a la base de datos el número de inventario del activo fijo borrado en la tabla para que sea eliminado de la base de datos y devuelve los valores actualizados.

Cada uno de estos métodos es capaz de detectar y capturar si ocurre una excepción, que puede ser generada por la no disponibilidad de la base de datos, formatos desconocidos o que no correspondan con el tipo de datos específico que se tiene que recibir, existencia en la base de datos del nuevo medio que se desea agregar, entre otros. Estas excepciones generadas son tratadas en la clase *ReglasNegocioException.cs* mostrando un mensaje al usuario donde se le informa el tipo de error ocurrido para que este pueda darle solución.

2.2.3 La capa de acceso a datos

En esta capa se definen las funciones correspondientes a la conexión del software con la base de datos y el pase de parámetros a esta. Las funciones de esta capa se encuentran en el diagrama de clases que se muestra a continuación.

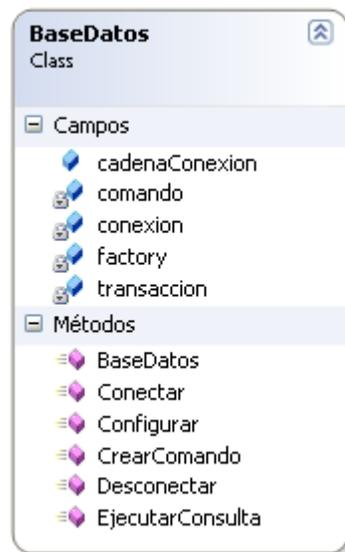


Fig. 2.8 Diagrama de clase de la capa de Acceso a Datos.

Los métodos fundamentales de esta clase son: *Configurar*, *Conectar*, *CrearComando* y *EjecutarConsulta*. A continuación se hace una descripción más detallada de los métodos:

- *Configurar*: se encarga de especificar los parámetros *Proveedor* y *Cadena_conexion*, los cuales especifican a que servidor de base de datos se va a conectar el software y quien es el proveedor de esta. Este método es llamado en el método *BaseDatos* que es el constructor de esta clase.
- *Conectar*: se encarga de establecer la conexión con la base de datos en el servidor, de no ser posible esto, se genera una excepción que es controlada por este método y que muestra al usuario un mensaje de error, si existe una conexión abierta, este método igualmente, controla esta excepción y muestra un mensaje indicando al usuario lo ocurrido.
- *CrearComando*: es el método que se encarga de convertir en sentencias SQL las acciones realizadas por el usuario y los parámetros a pasar a la base de datos, para que puedan ser ejecutadas en los procedimientos almacenados.
- *EjecutarConsulta*: ejecuta el comando SQL en la base de datos y obtiene los datos que esta devuelve acorde a la acción realizada.
- *Desconectar*: terminada la operación sobre la base de datos la conexión es cerrada por este método.

Si existe algún problema y la conexión no puede ser establecida se genera una excepción que es tratada en la clase *AccesoDatosException* y se le muestra al usuario un mensaje indicándole cual fue el tipo de error ocurrido.

2.3 Código de barra seleccionado.

Para la entrada y codificación de los datos se propone utilizar el código de barras 128, que es una variante del código 39, pero este permite mayor cantidad de información codificada. Puede codificar los 128 caracteres ASCII.



Código 128.

Fig. 2.9 Ejemplo de código 128.

El código 128 utiliza 4 diferentes grosores para las barras y los espacios y tiene una densidad muy alta y es utilizado en inventarios y control de mercadería en grandes almacenes, por lo que se ajusta bien a nuestro diseño.

2.4 Control automatizado de los activos fijos tangibles.

El proceso automatizado de control de activos fijos consistiría en hacer un recorrido, por los distintos locales a controlar, con el scanner de códigos de barras y el hardware (todavía sin construir) asociado a este. Se van leyendo los códigos en cada uno de los medios, estos códigos se almacenan en una pequeña memoria que posee este hardware y al terminar el recorrido son descargados en la PC mediante el software elaborado. Estos datos obtenidos son sometidos a un proceso de chequeo contra los que existen en la base de datos para encontrar si falta algún activo fijo en los locales controlados.

CAPÍTULO 3. MANUAL DE USUARIO Y PRUEBAS AL SISTEMA.

Este capítulo está dedicado a la explicación de la forma de manejo del software y algunas pruebas realizadas al mismo.

3.1 Manual de usuario y pruebas al sistema

Luego de realizado el software es necesario dar una breve explicación de cómo trabajar con él, de manera tal, que si el usuario no está familiarizado con su manejo, en poco tiempo logre dominarlo, logrando así un mejor desempeño del mismo y ahorro de tiempo a la hora de procesar los datos.

El programa permite interactuar con los datos de los Activos Fijos Tangibles de una empresa o entidad, almacenados en una Base de Datos, desde esta interfaz se pueden consultar los datos, realizar búsquedas y agregar o borrar medios, así como modificar los ya existentes.

Al iniciar, el programa muestra una lista (1) con los datos de los Activos Fijos Tangibles, los cuáles están almacenados en una base de datos en un servidor, que puede ser la misma computadora donde se está trabajando o una en cualquier otro lugar de la red.

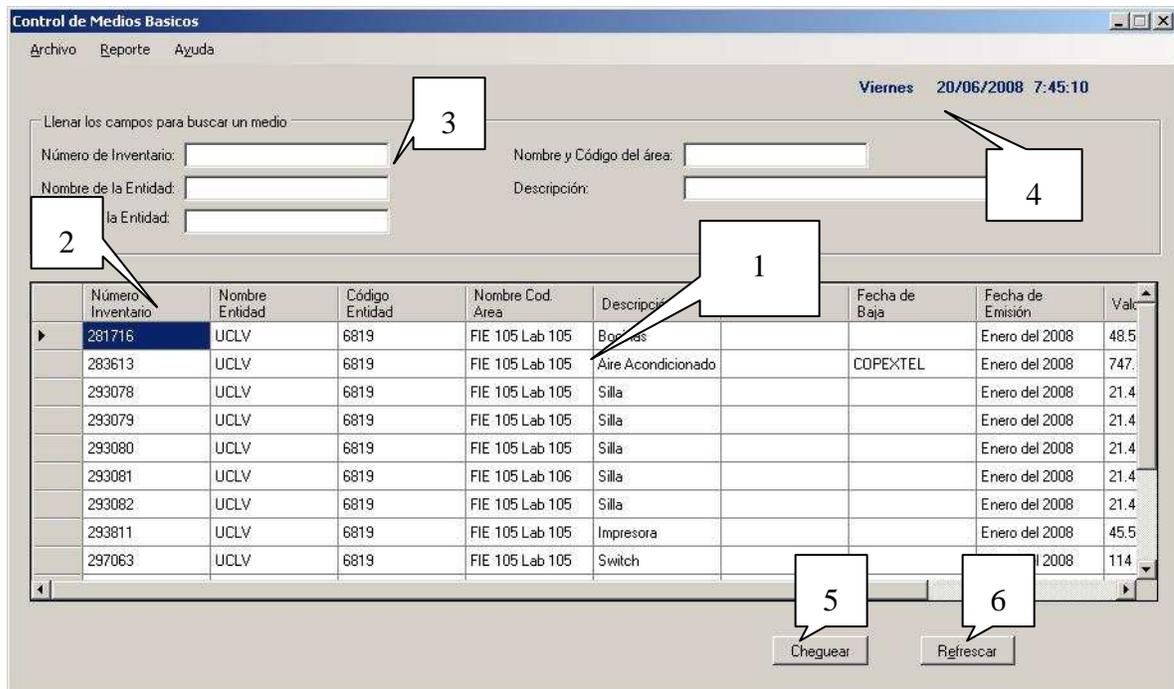


Fig. 3.1 Imagen del programa al iniciar.

La tabla se puede organizar de distintas formas haciendo clic sobre el encabezado (2) de cada columna se puede organizar de mayor a menor o alfabéticamente en dependencia de la columna que se este ordenando. Así se puede organizar de menor a mayor por los Números de Inventario o alfabéticamente por la Descripción.

En la parte superior aparecen cinco cajas de texto (3), mediante las cuales se puede realizar un filtrado a los datos mostrados en la tabla ya sea por su Número de Inventario, Nombre de la Entidad o Descripción, también se puede hacer un filtrado múltiple, o sea, con mas de un campo a la vez.

A continuación se muestra una imagen de un filtrado múltiple como ejemplo a lo explicado anteriormente, se realiza el filtrado teniendo en cuenta todos los equipos que tengan un Número de Inventario que comience con 28 y en la descripción diga unidad.

The screenshot shows a web application window titled "Control de Medios Basicos". At the top right, it displays the date and time: "Viernes 20/06/2008 7:46:57". Below this is a search section with the instruction "Llenar los campos para buscar un medio:". There are four input fields: "Número de Inventario" (containing "28"), "Nombre y Código del área:", "Nombre de la Entidad:", and "Descripción" (containing "unidad"). The "Número de Inventario" and "Descripción" fields are circled in red. Below the search section is a table with the following data:

	Número Inventario	Nombre Entidad	Código Entidad	Nombre Cod. Área	Descripción	Fecha de Alta	Fecha de Baja	Fecha de Emisión	Valor C
▶	281716	UCLV	6819	FIE 105 Lab 105	Bocinas			Enero del 2008	48.52
	283613	UCLV	6819	FIE 105 Lab 105	Aire Acondicionado		COPEXTEL	Enero del 2008	747.5
	297910	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	1083.23
	297911	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	1083.23
	297912	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	1083.23
	297913	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	1083.23
*									

At the bottom of the interface, there are two buttons: "Chegurar" and "Refrescar".

Fig. 3.2 Ejemplo de filtrado múltiple.

En la parte superior derecha se muestra la fecha actual y la hora (4) para comodidad del usuario.

Para borrar un medio de la Base de Datos se selecciona la fila completa haciendo clic en el encabezado de la misma y se presiona la tecla "Delete" y en el cuadro de dialogo que se muestra se presiona el botón "Yes".

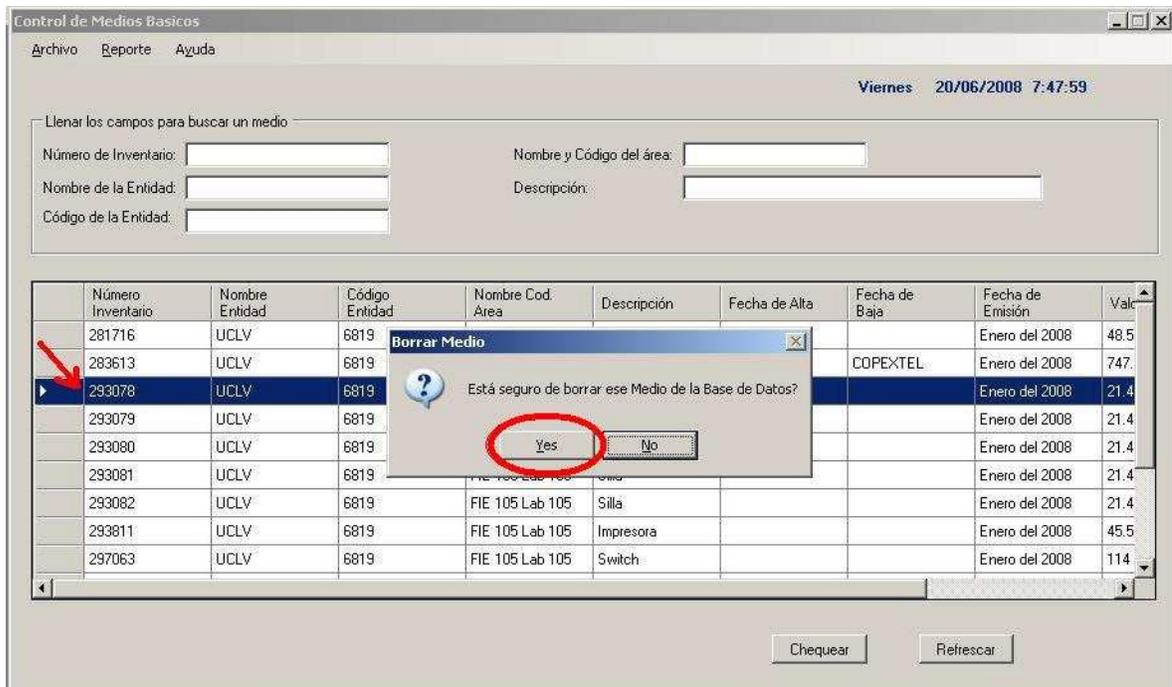


Fig. 3.3 Borrar un medio de la Base de Datos.

Si se desea ingresar un nuevo medio a la Base de Datos, se hace doble clic en la primera celda de la última fila, que esta en blanco, y se introduce primero su Número de Inventario. Siempre el Número de Inventario debe ser lo primero en introducirse porque la Base de Datos no acepta este parámetro como nulo, por lo tanto si se deja en blanco devuelve un mensaje de error.

Número Inventario	Nombre Entidad	Código Entidad	Nombre Cod. Área	Descripción	Fecha de Alta	Fecha de Baja	Fecha de Emisión	Val
293082	UCLV	6819	FIE 105 Lab 105	Silla			Enero del 2008	21.4
293811	UCLV	6819	FIE 105 Lab 105	Impresora			Enero del 2008	45.5
297063	UCLV	6819	FIE 105 Lab 105	Switch			Enero del 2008	114
297910	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	108
297911	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	108
297912	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	108
297913	UCLV	6819	FIE 105 Lab 105	Unidad Central			Enero del 2008	108
298563								

Fig. 3.4 Insertando un nuevo medio en la Base de Datos.

En la parte inferior derecha aparecen dos botones, *Chequear* (5) y *Refrescar* (6). El botón *Refrescar* es para actualizar los datos mostrados en la tabla desde la Base de Datos, mientras que el botón *Chequear* es para comparar los códigos previamente escaneados y almacenados con los que se tienen en la Base de Datos, en caso de de que algún código de los existentes no aparezca entre los escaneados se marcara en rojo ese fila, indicando que ese medio no fue encontrado en el local donde se realizó el chequeo. También se puede hacer un chequeo de los medios junto con un filtrado, para facilitar y especificar los medios de interés.

Por ejemplo se van a chequear los códigos que pertenezcan al área con nombre y código *FIE 105 Lab 105* y se marcarán en rojo las filas de los medios que no se encontraron en ese local. Haciendo clic en el botón chequear aparece una nueva ventana en la que se le pregunta al usuario de que forma va a entrar los códigos obtenidos en el local. Esto puede hacerse de 3 formas:

- **Descargar los códigos directamente del scanner:** se selecciona la primera opción y se presiona OK.

- **Los códigos están en un archivo predeterminado:** seleccionando la segunda opción y haciendo clic en el botón se especifica la ubicación donde se encuentra el archivo.
- **Los códigos va a ser entrados manualmente:** marcando la tercera opción se activa una caja de texto la que le permite al usuario introducir los códigos en el momento.

En este caso los códigos están almacenados en un archivo predeterminado y se van a leer de ahí:

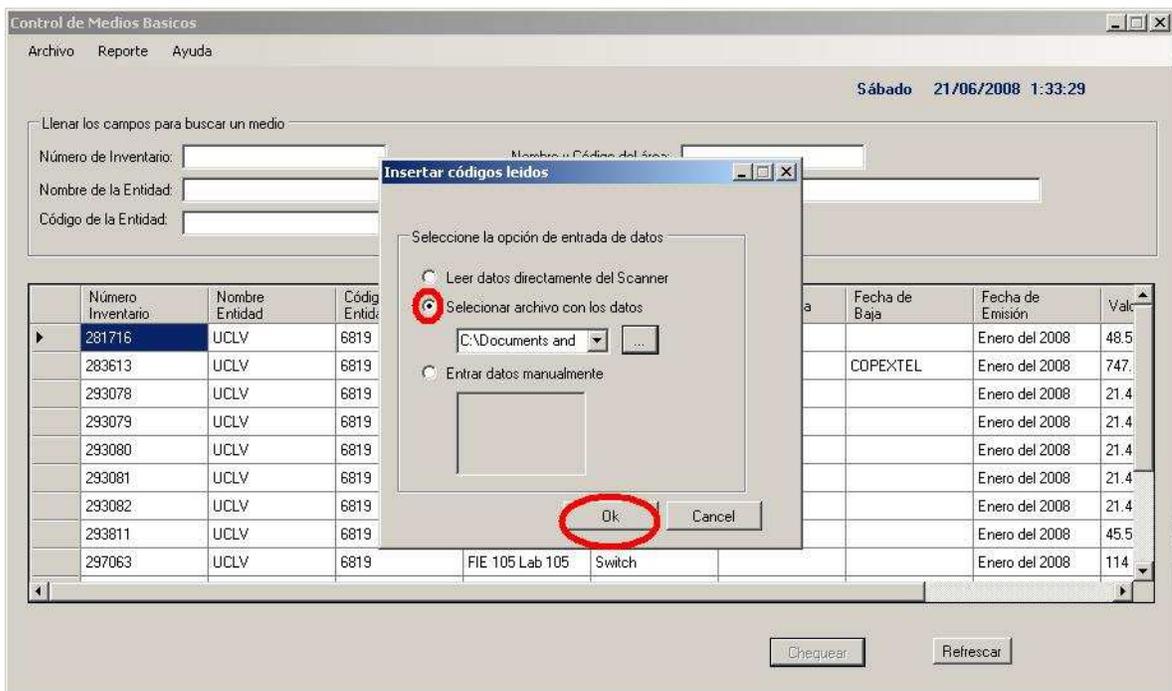


Fig. 3.5 Selección del origen de los datos a chequear.

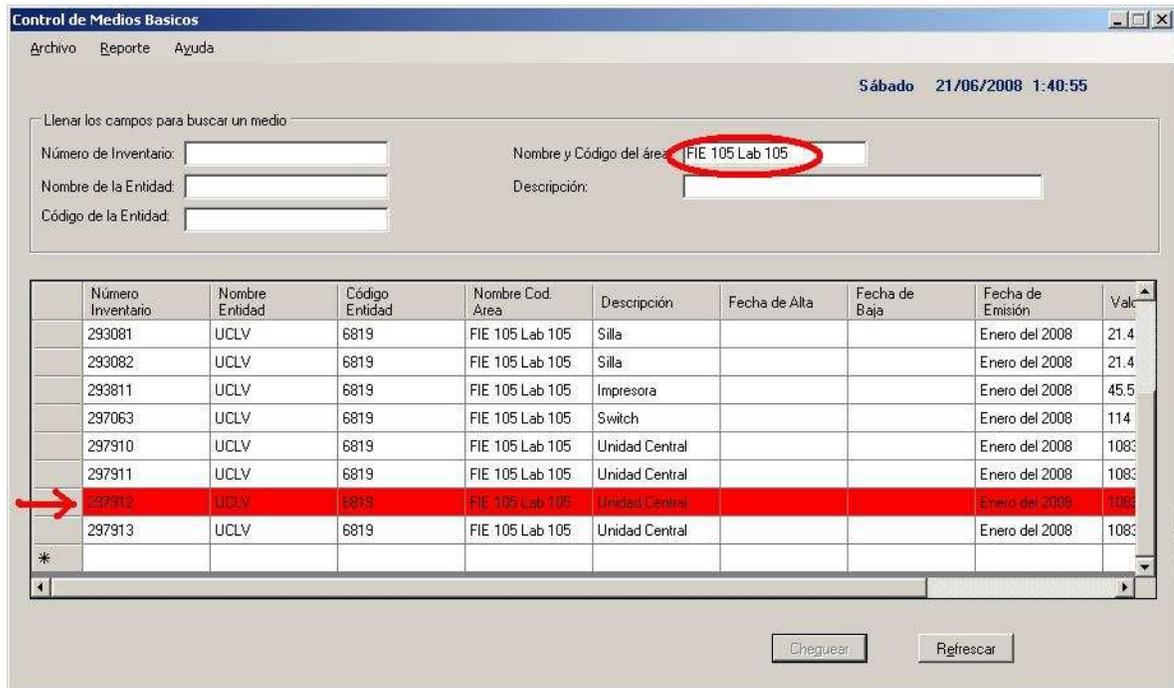


Fig. 3.6 Resultado del chequeo desde archivo.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Con la realización de este trabajo se logró realizar un software para el control de activos fijos tangibles en la facultad de Ingeniería Eléctrica utilizando el C# 2.0 y el SQL Server 2005.

Este software es ejecutable en Windows y posee estructura que facilita al usuario un fácil y rápido manejo de los datos para el control de activos fijos, posibilitando que sea extensible a otras áreas de la UCLV.

Los resultados obtenidos durante el proceso de desarrollo de este trabajo demuestran que:

- Con la realización de un software de este tipo se mejora la calidad del proceso de control de activos fijos, haciéndolo más rápido y potente contra errores humanos.
- El estado del arte sobre la identificación por códigos de barras demuestra que es en estos momentos uno de los métodos más utilizados mundialmente en esta rama y está siendo explotado en casi todos los países por las medianas y grandes empresas.
- Los software pertenecientes a la plataforma .NET como el C# son una de las herramientas más potentes para la realización de aplicaciones, al igual que el SQL Server como gestor de bases de datos.

Recomendaciones

Se recomienda la realización y creación del hardware para comunicar el scanner de códigos de barra con la PC, permitiendo que se pueda llevar a cada uno de los lugares en los que se van a controlar los activos fijos, este hardware también debe dar la posibilidad al scanner de ser independiente de fuentes fijas de alimentación y poseer una memoria para almacenar los datos recopilados.

REFERENCIAS BIBLIOGRÁFICAS

[1] Cardona, Verónica: “Código de Barras”.

URL: <http://html.rincondelvago.com/codigo-de-barras-html>

[2] María, Yenina:”Acerca de los códigos de barras...”

URL: www.monografias.com

[3] Wikimedia Foundation, Inc. 2007, Código de barras.

URL: http://es.wikipedia.org/wiki/C%C3%B3digo_de_barras

[4] Manual de usuario EAN.UCC.

URL: www.eanchile.cl/ean_manuales/manual_del_usuario_mar03.pdf

[5]

[6]

[7] Seco, J. A. G. (2001). El Lenguaje de Programación C#.

[8] Liberty, J. (2001). Programming C# (First Edition ed.).

[9] Kimmel, P. (2000). Advanced C# Programming.

[10] Ivar Jacobson, G. B., James Rumbaugh. (2000). El Proceso Unificado de Desarrollo de Software.

[11] Jacobson, I. (1998). Applying UML in The Unified Process. Presentación. Rational Software.

BIBLIOGRAFÍA.

- Cota, A. (1994). Ingeniería de Software. Soluciones Avanzadas. pp. 5-13.
- Archer, T. (2001). A Fondo C#.
- Moratalla, J. (2001). Bases de Datos con SQL Server 2000. Transact SQL
- Lewis, G. (1994). What is Software Engineering?. DataPro (4015). pp. 1-10.
- Larman, C. (1999). UML y Patrones: Pearson.
- Anónimo: CODIGO UCC/EAN-128. Artículo de EAN-Venezuela. URL:
www.eanve.org
- Anónimo: ABC del código de barras. Artículo publicado en internet. URL:
www.arzp.com/bar=code/lectores.html
- Azalea Software, Inc. 2007, Barcode Software & Bar Code Fonts. URL:
www.azalea.com
- Sofía, Marcelo: Guía de referencia EAN/UCC-14. Artículo de la firma EAN Argentina-código.
- Anónimo: Uso de Códigos de Barras para Principiantes. Artículo de la firma Symbol Technologies. Inc.

ANEXOS

Anexo I El Código de Barras.

Concepto.

Después de varias consultas bibliográficas acerca de la definición del Código de Barras, se asume el concepto de que este es una disposición en paralelo de barras y espacios que contienen información codificada en barras y espacios del símbolo.

Los Códigos de Barras se asemejan a la versión impresa del código Morse, sólo que en este caso, se representan los puntos con barras angostas, y las rayas con barras anchas.

Los Códigos de Barras representan un método simple y fácil para codificar información de texto, que puede ser leída por dispositivos ópticos, y enviada a la computadora como si hubiera sido tecleada.

Principio de funcionamiento del Código de Barras

Los Códigos de Barras basan su funcionamiento en el efecto físico óptico de reflexión y absorción de la luz. Los códigos de barras se leen pasando un pequeño haz de luz sobre el símbolo del código de barras impreso. La secuencia de barras oscuras y blancas absorben y reflejan la luz emitida por el escáner de forma tal que se genere una señal eléctrica que sea reflejo del patrón de barras. El escáner "lee" los códigos de barras, fijándose tanto en las líneas negras, como en el tamaño y espacios en blanco entre las barras.

Los escáner necesitan de un decodificador para traducir el símbolo leído, este puede estar integrado al escáner o exterior al mismo.

Características del Código de Barras.

Entre las principales características que definen a una simbología de Código de Barras pueden encontrarse las siguientes:

- **Densidad:** es la anchura del elemento (barra o espacio) más angosto dentro del símbolo del Código de Barras. Esta dado en miles (milésimas de pulgada). Un código de barras no se mide por su longitud física sino por su densidad.
- **WNR:** (Del ingles: Wide to Narrow Ratio): es la razón del grosor del elemento más angosto contra el más ancho. Usualmente es 1:3 o 1:2.
- **Quiet Zone (Zona blanca):** es el área blanca al principio y al final de un símbolo de código de barras. Esta área es necesaria para una lectura conveniente del símbolo.
- **Numéricas o alfanuméricas:** dependen del tipo de información que se incluya en el código, si son solo numérico o si incluyen caracteres.

Código de Barras lineales y su utilización.

Universal Product Code (U.P.C.)

UPC es la simbología más usada en el comercio minorista de EEUU, pudiendo codificar solo números. El estándar UPC (denominado UPC-A) es un número de 12 dígitos. El primero es llamado “número del sistema”. La mayoría de los productos tienen un “1” o un “7” en esta posición. Esto indica que el producto tiene un tamaño y peso determinado, y no un peso variable. Los dígitos del segundo al sexto representan el número del fabricante. Esta clave de 5 dígitos (adicionalmente al número del sistema) es única para cada fabricante, y la asigna un organismo rector evitando códigos duplicados. Los caracteres del séptimo al onceavo son un código que el fabricante asigna a cada uno de sus productos, denominado “número del producto”. El doceavo carácter es el “dígito verificador”,

resultando de un algoritmo que involucra a los 11 números previos. Existen variantes del UPC.



Fig.1 Código de Barras UPC-A

European Article Numbering (E.A.N.)

El sistema de codificación EAN es usado tanto en supermercados como en comercios. Es un estándar internacional, creado en Europa y goza de gran aceptación mundial. Identifica a los productos comerciales por medio del Código de Barras, indicando el país-empresa-producto con una clave única internacional. Hoy en día es casi un requisito indispensable tanto para el mercado interno como internacional.



Fig. 2 Código de Barras EAN-13

Código 39

Es un estándar no utilizado para la industria alimenticia. Generalmente se utiliza para identificar inventarios y para propósitos de seguimiento en las industrias, esta simbología es actualmente la más usada para aplicaciones industriales y comerciales para uso interno ya que se permite la codificación de caracteres numéricos, letras mayúsculas y algunos

símbolos como -, \$, /, +, % y “espacio”. Se utilizan solo dos grosores tanto para las barras como para espacios. Sin embargo el código 39 produce una barra relativamente larga y puede no ser adecuada si la longitud es un factor de consideración.



Fig. 3 Código 39.

Código 128

Utilizado cuando es necesaria una amplia selección de caracteres más de lo que puede proporcionar el código 39. El código 128 utiliza 4 diferentes grosores para las barras y los espacios y tiene una densidad muy alta, ocupando en promedio solo el 60% del espacio requerido para codificar información similar en Código 39. Puede codificar los 128 caracteres ASCII. Cuando la dimensión de la etiqueta es importante, el Código 128 es una buena alternativa porque es muy compacta lo que resulta en un símbolo muy denso. Esta simbología se usa a menudo en la industria de envíos donde el tamaño de la etiqueta es importante.



Fig. 4 Código 128.

1.5. Código de Barras de dos dimensiones.

Los datos están codificados en la altura y longitud del símbolo, y en éstos códigos la información no se reduce sólo a un código lineal como el código de un artículo, sino que puede almacenar gran cantidad de datos.

La principal ventaja de utilizar códigos de 2 dimensiones es que este contiene una gran cantidad de información que puede ser leída de manera rápida y confiable, sin necesidad de

acceder a una base de datos en donde se almacene dicha información (el caso de los códigos de 1 dimensión)

La seguridad que es capaz de incorporarse en estos códigos, los hace casi invulnerables a un sabotaje. Para estropear la legibilidad de un código unidimensional, basta con agregar otra barra al inicio o final del símbolo o trazar una línea paralela a las barras en cualquier lugar dentro del código. Los códigos de 2D se pueden construir con muchos grados de redundancia, duplicando así la información en su totalidad o sólo los datos vitales. La redundancia aumenta las dimensiones del símbolo, pero la seguridad del contenido se incrementa notablemente.

Se han hecho pruebas de resistencia a códigos bidimensionales perforándolos, marcándolos con tinta y maltratándolos. El símbolo es legible aún después de todos estos abusos.

Los códigos de 2D deben ser considerados como un complemento a la tecnología tradicional de códigos de 1D, no como su reemplazo; y las ventajas deben ser comparadas contra el incremento en costo.

Ventajas del Código de Barras.

Para identificar objetos y facilitar el ingreso de información, eliminando la posibilidad de error en la captura, ha sido creado el Código de Barras, el cual tiene ventajas sobre los otros procedimientos de colección de datos como:

- Su impresión es a bajos costos.
- Permite porcentajes muy bajos de error.
- Facilita rapidez en la captura de datos.
- Los equipos de lectura e impresión son flexibles y fáciles de conectar e instalar.

Anexo II Modelo para el control de activos fijos tangibles.

Nombre de la Entidad		UCLV		Código de la Entidad		6819	
Nombre y Código del Área			Fac. Ing. Eléctrica				
Fecha de emisión o actualización			Enero del 2008				
No de Inventario	Descripción del AFT			Fecha de Alta	Valor en MN	Valor en MLC	Fecha de Baja AFT
Firma de la persona que elabora el modelo				Firma de la persona del área		Firma del controlador en Contabilidad	
							No consecutivo

SC-1-07