



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

Universidad Central "Marta Abreu" de Las Villas

Facultad de Ingeniería Eléctrica

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

TRABAJO DE DIPLOMA

**Sistema Automatizado para el Control de
Asistencia de los Trabajadores**

Autor: Dairo Blanco Guevara.

Tutores: Ing. Alexander Falcón Ruiz.

Ing. Denis Hernández Pacheco.

Santa 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 Asistencia de los Trabajadores

Autor: Dairo Blanco Guevara.

dairo@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

PENSAMIENTO

Inteligencia es la habilidad de adaptarse a los cambios.

Stephen Hawking

Dedicatoria

DEDICATORIA

A mis padres que me lo han dado todo,

A mi hermano y mis amigos,

A mi novia y mi hermosa familia.

Agradecimientos



AGRADECIMIENTOS

A mis tutores Alexander y Denis (Ramón) por la ayuda incondicional,

Al excelente claustro de profesores de la FIE.

A todos lo que de una forma u otra han hecho posible la realización de este trabajo,

A mis padres por su excelente educación y por su sacrificio en todo momento,

A mi hermano, a mi familia, a mi novia y a mis amigos, por la confianza depositada en mí.

Tarea Técnica

TAREA TÉCNICA

- La realización de un estudio de factibilidad del sistema de registro de asistencia para trabajadores.
- La realización de un estudio de sistemas similares desarrollados en la UCLV.
- La realización de un estudio del lenguaje de programación C# y del uso del ambiente de programación Visual Studio .NET 2005.
- La realización de un estudio del diseño de bases de datos y del Sistema de Gestión de Bases de Datos (SGBD) MS SQL Server 2005.
- La realización de entrevista a especialistas, estudio de los modelos utilizados en el proceso y del tratamiento de los datos.
- La realización de un software para el registro y control de asistencia de los trabajadores.
- La realización de un manual de usuario.

Firma del Autor

Firma del Tutor

Resumen

RESUMEN

En el presente trabajo de diploma se describe el desarrollo de un software para controlar la asistencia de los trabajadores del Centro de Información Científico-Técnica (CDICT) de nuestra universidad. El objetivo es reemplazar el sistema manual que existe actualmente en este lugar por un sistema informatizado que realice el proceso de manera más rápida, confiable y económica. Dicho sistema permite procesar los datos registrados y obtener reportes de utilidad.

Para identificar a los trabajadores, el sistema toma como entrada el código de barras impreso en los carnés que los identifican, empleando para ello un escáner de código de barras conectado a una PC. En la programación de la aplicación de software se emplea el lenguaje de programación C# 2.0 y el ambiente de desarrollo integrado MS Visual Studio .NET 2005 Enterprise Edition. En el desarrollo de la base de datos se emplea el SGBD MS SQL Server 2005 Enterprise Edition.

Tabla de Contenidos

TABLA DE CONTENIDOS

PENSAMIENTO	ii
DEDICATORIA.....	iv
AGRADECIMIENTOS.....	vi
TAREA TÉCNICA.....	viii
RESUMEN.....	x
INTRODUCCIÓN.....	2
CAPÍTULO 1. INTRODUCCIÓN AL PROYECTO.....	6
1.1 Descripción del proyecto.....	6
1.2 Códigos de barras.....	7
1.2.1 Concepto.....	8
1.2.2 Principio de funcionamiento.....	8
1.2.3 Características principales de un código de barras.....	8
1.2.4 Aplicaciones.....	9
1.3 SQL Server 2005.....	9
1.4 C Sharp.....	11
1.4.1 Características.....	11
1.4.2 Ventajas de C#.....	12
1.5 Descripción del diseño arquitectónico del sistema de información.....	12
CAPÍTULO 2 Descripción del Sistema Automatizado Control de Asistencia.....	15
2.1 Selección de las herramientas.....	15
2.2 Diseño de la base de datos.....	15
2.2.1 Diseño conceptual.....	16

2.2.2 Diseño lógico.....	18
2.2.3 Diseño físico.....	21
2.3 Modelo de tres capas aplicado al software.....	22
2.3.1 La capa de acceso a datos.....	23
2.3.2 La capa de negocio.....	24
2.3.3 La capa de presentación o aplicación.....	26
CAPÍTULO 3 MANUAL DE USUARIO.....	30
CONCLUSIONES.....	36
RECOMENDACIONES.....	38
REFERENCIAS BIBLIOGRÁFICAS.....	40
ANEXO I Hoja de Firmas.....	43
ANEXO II Modelo RH-3.....	44

Introducción

INTRODUCCIÓN

En la mayor parte de los centros laborales del mundo se controla de una forma u otra la asistencia de sus trabajadores. Esta tarea, que anteriormente se realizaba manualmente, con el llenado de tarjetas por los mismos trabajadores, ha experimentado en los últimos años grandes cambios. Éstos se han debido, en gran medida, al uso de lenguajes de programación y al desarrollo de las bases de datos para el almacenamiento de la información.

En nuestro país se está experimentando un avance en cuanto a la informatización de los controles de acceso, y así mismo los controles de asistencia en las empresas con la elaboración de algún software con estos fines.

Los códigos de barras, son una técnica de entrada de datos, con imágenes formadas por combinaciones de barras y espacios paralelos, de anchos variables. Algunas de las aplicaciones de los códigos de barras son el control de mercancía, control de inventarios, control de tiempo y asistencia, identificación de paquetes y embarques y recibos.

Actualmente en la mayoría de los centros laborales del país no se utiliza los códigos de barras como identificación personal para el control de la asistencia de sus trabajadores. Téngase en cuenta que el uso de los mismos requiere de algunas inversiones tanto en hardware como en software.

Con la realización de este trabajo se persigue mejorar el proceso de control de asistencia de los trabajadores del CDICT, de manera que se haga más eficaz y robusto a posibles errores, utilizando la identificación por códigos de barras. Se pretende además que esta solución

pueda ser extendida a otros centros laborales del país que no cuentan con automatización para dicho proceso.

Objetivo:

El objetivo que se persigue con el trabajo es desarrollar un software para llevar el control de la asistencia de los trabajadores del CDICT y confeccionar reportes con estos datos, teniendo éste una interfaz cómoda, amigable, fácilmente extensible, con código ejecutable para MS Windows, utilizando el lenguaje de programación C# 2.0 y el SGBD MS SQL Server 2005 Enterprise Edition.

Objetivos específicos:

- La realización de un estudio de factibilidad del sistema de registro de asistencia para trabajadores.
- La realización de un estudio del lenguaje de programación C# 2.0, del diseño de bases de datos y del SGBD MS SQL Server 2005.
- La realización de entrevistas a especialistas, estudio de los modelos utilizados en el proceso y del tratamiento de los datos.
- La realización de un software y de una base de datos para el registro y control de asistencia de los trabajadores.
- La realización de un manual de usuario.

Hipótesis de la investigación.

Si se elabora un software para el control de la asistencia de los trabajadores del CDICT, 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 así como los requisitos de instalación del software.

CAPÍTULO 1. INTRODUCCIÓN AL PROYECTO

1.1 Descripción del proyecto.

Generalmente las empresas requieren contar con herramientas de gestión para controlar y cuantificar los distintos aspectos de su operación. Aplicaciones para control de almacenes, inventarios, planillas, ventas, cuentas por cobrar y pagar, contabilidad, entre muchos otros, son usuales en la gran mayoría de empresas, independientemente de su tamaño o giro de negocio.

Una de las áreas que más a tardado en automatizarse ha sido la del Control de Asistencia y/o permanencia del personal, debido en gran medida por la aún amplia utilización de sistemas manuales, de control basados en los antiguos relojes mecánicos que utilizan tarjetas de cartón, o únicamente con partes de asistencia que no son otra cosa que hojas de papel con líneas numeradas en las que los empleados firman e indican la hora en la cual ingresan o salen de la empresa.

Actualmente el proceso en el CDICT de la Universidad Central “Marta Abreu” de Las Villas presenta algunas dificultades, el trabajador tiene que llenar los campos de hora de entrada y de salida de forma manual, con los inconvenientes asociados a errores humanos (ilegibilidad, llenado erróneo de otros campos, olvido de llenar algún campo, fraude, entre otros) que esta operación acarrea. Debe haber una persona responsable del registro de manera que no se produzcan alteraciones en el mismo. Existe un gasto de papel y de tonel de impresora, para imprimir modelos diariamente. El tiempo de procesamiento de los modelos asociados al proceso (tanto en cada área como por el personal de recursos humanos) es largo y pueden aparecer errores asociados a la intervención humana.

Con la ayuda del software se podrían mejorar considerablemente estas dificultades. El procesamiento sería más rápido y eficaz al realizarse de forma automatizada, se eliminarían los problemas asociados con errores humanos, el ahorro de insumos (papel y tonel) para la impresión de los modelos diariamente se reduciría totalmente. Además se cuenta con los diferentes medios para la realización del software, tales como computadora, escáner, documentación sobre las herramientas a utilizar e información transmitida por los especialistas.

El presente proyecto consiste en un software para la automatización del procesamiento de la asistencia de los trabajadores del CDICT. Cada uno de los trabajadores del centro posee un carné con un código de barras que lo identifica, esta característica será aprovechada por el sistema como entrada de datos usando un lector de códigos de barras. Los datos se guardarán en una base de datos administrada por el SGBD SQL Server 2005. La aplicación que manipula estos datos y que brinda una interface visual al operador se confeccionará empleando el lenguaje de programación C# 2.0 y el ambiente de desarrollo integrado Visual Studio .NET 2005. La aplicación es diseñada basada el modelo de tres capas que se explicará más adelante. A continuación se dará una breve panorámica acerca de las herramientas utilizadas para la confección del software.

1.2 Código de barras.

Los códigos de barras se han integrado en cada aspecto de nuestras vidas, se localizan en el supermercado, en tiendas departamentales, farmacias, etc. Han sido aceptados como parte de nuestra vida diaria, pero a veces no sabemos qué es lo que representan.

No se requiere de gran conocimiento técnico para entenderlos, los códigos de barras son solo una forma diferente de codificar números y letras usando una combinación de barras y espacios en diferentes medidas. Es otra forma de escritura, ya que reemplazan el tecleo de datos para recolectar información.

Los códigos de barras son una forma fácil, rápida y precisa de codificar información.

1.2.1 Concepto

El Código de Barras es una disposición en paralelo de barras y espacios que contienen información codificada en las barras y espacios del símbolo, almacena información, datos que pueden ser reunidos en él de manera rápida y con una gran precisión. Representan un método simple y fácil para codificación de información de texto que puede ser leída por dispositivos ópticos, los cuales envían dicha información a una computadora como si la información hubiese sido tecleada ^{[1][2]}.

1.2.2 Principio de funcionamiento

El funcionamiento de los códigos de barras se basa en el efecto físico óptico de reflexión y absorción de la luz. Se leen pasando un pequeño haz de luz sobre el símbolo impreso ^[3]. 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 al símbolo leído, este puede estar integrado al escáner o exterior al mismo ^[4].

Para el trabajo con los códigos de barras existen comisiones a nivel mundial y podemos destacar, dentro de la amplia bibliografía que existe sobre el tema, el "Manual Mundial del Usuario EAN, UCC" ^[5].

1.2.3 Características principales de un código de barras ^{[6][4][7]}.

Las principales características que definen a una simbología de código de barras son las siguientes:

- Numéricas o alfanuméricas
- De longitud fija o de longitud variable
- Discretas o continuas
- Número de anchos de elementos
- *Quiet Zone* (es el área blanca al principio y al final de un símbolo del código de barras)

1.2.4 Aplicaciones.

Las aplicaciones del código de barras cubren prácticamente cualquier tipo de actividad humana, tanto en industria, comercio, instituciones educativas, instituciones médicas, gobierno, etc. Cualquier negocio se puede beneficiar con la tecnología de captura de datos por código de barras, tanto el que fábrica, como el que mueve, como el que comercializa ^[8] ^[9].

Entre las aplicaciones que tiene podemos mencionar:

- Control de inventario
- Control de tiempo y asistencia
- Control de acceso
- Punto de venta
- Control de calidad
- Control de embarques y recibos

1.3 SQL Server 2005 ^{[10][11]}.

Hoy en día las organizaciones enfrentan numerosos desafíos de datos, tales como la necesidad de tomar decisiones más rápidas y más orientadas a datos, la necesidad de aumentar la productividad y flexibilidad del personal de desarrollo y presionan para reducir los presupuestos generales de informática (IT) a la vez que escalan la infraestructura para satisfacer las exigencias cada vez mayores.

SQL Server es una solución de datos global, integrada y de extremo a extremo que habilita a los usuarios en toda su organización mediante una plataforma más segura, confiable y productiva para datos empresariales. SQL Server 2005 provee herramientas sólidas y conocidas a los profesionales de IT, así como también a trabajadores de la información, reduciendo la complejidad de la creación, despliegue, administración y uso de aplicaciones analíticas. La figura 1.1 muestra el diseño de la plataforma de datos SQL Server 2005.

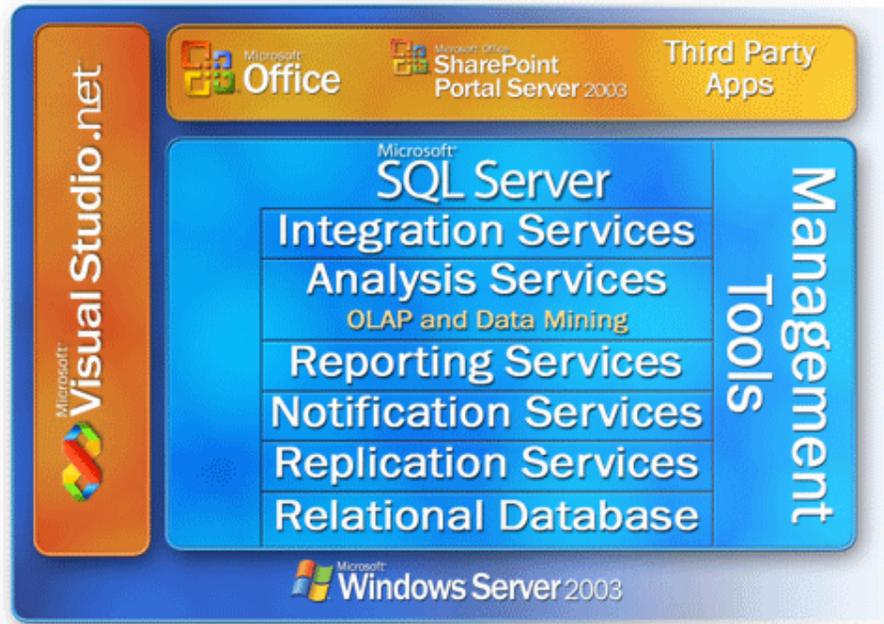


Figura 1.1 Diseño de la plataforma de datos SQL Server 2005.

La plataforma de datos SQL Server incluye las siguientes herramientas:

- **Base de datos relacional.** Un motor de base de datos relacional más segura, confiable, escalable y altamente disponible con mejor rendimiento y compatible para datos estructurados y sin estructura (XML).
- **Reporting Services.** Una solución global para crear, administrar y proporcionar tanto informes tradicionales orientados al papel como informes interactivos basados en la Web.
- **Herramientas de administración.** SQL Server incluye herramientas integradas de administración para administración y optimización avanzadas de bases de datos, así como también integración directa con herramientas tales como Microsoft Operations Manager (MOM) y Microsoft Systems Management Server (SMS).

SQL Server 2005 constituye una opción más avanzada que su versión precedente integrando herramientas de administración y con potencialidades de interacción con Visual Studio 2005 y Microsoft Office entre otros.

1.4 C SHARP ^[23].

Son muchos los lenguajes de programación que han aparecido a lo largo de la historia de las computadoras y a pesar de que con cada uno se pueden lograr hacer grandes aplicaciones, como todo, tienen sus ventajas y desventajas. Una de las desventajas de estos programas es que entre ellos existen maneras muy diferentes de estructurar el código, además de que cada programa maneja sus propias librerías y sintaxis ^[12].

Por todos estos aspectos la empresa de Microsoft creó 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: C#.

1.4.1 Características ^{[13][14]}.

Facilidad de uso: el ambiente de trabajo es muy cómodo ya que tiene un ambiente amigable y clásico de las aplicaciones de Windows. En cuanto a la forma de programar, será fácil de usar para quien está familiarizado con C++ o Java.

Programación orientada a objetos: esta forma de programación ahorra mucho código, lo cual indica que partes de código son reutilizables para no volverlas a escribir, con lo cual se afirma que C# presenta las características necesarias para considerarlo como un lenguaje orientado a objetos, tales son: encapsulación, herencia y polimorfismo. Además una de las mejoras que presenta este lenguaje con respecto a este tipo de programación es que para evitar confusiones no existen variables o funciones globales, sino que se definen dentro de los tipos de datos.

Compatible: C# no sólo mantiene una sintaxis muy similar a C, C++ o Java sino que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes ^[15].

1.4.2 Ventajas de C#

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

- Declaraciones en el espacio de nombres: al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.
- Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- Atributos: cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.
- Métodos virtuales y redefiniciones: antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra *override*.

1.5 Descripción de diseño arquitectónico del sistema de información ^{[16][17]}.

La programación en múltiples capas o *multitier* es la técnica más efectiva en aplicaciones empresariales, debido a la fácil administración que implica el dividir los componentes de la aplicación en capas y la rapidez que esto trae en el desarrollo. Esta arquitectura consiste en dividir los componentes primarios de la aplicación, programarlos por separado y después unirlos en tiempo de ejecución.

Estas tres capas las podemos denominar como: Presentación, Reglas del Negocio y Acceso a Datos. A continuación se explica cada una:

- Presentación: En esta capa se diseña todo lo que constituye la interfaz gráfica y la interacción del usuario con el software.
- Reglas del Negocio: Son todas las subrutinas creadas con el propósito de regular alguna acción del usuario. Por ejemplo, en una aplicación bancaria una regla del negocio podría ser que el cliente no debe retirar por taquilla más de determinada cantidad de dinero y en caso de una petición de este tipo se genere un error.
- Acceso a datos: En ésta se programa todo lo que tiene que ver con el acceso a la base de datos. Esta capa queda encargada de tomar la información de la base de

datos dada una petición de la capa de Reglas del Negocio, que a su vez es generada por la capa de presentación.

El Sistema de Control de Asistencia para trabajadores es un sistema típico de información que incluye una interfaz gráfica de usuario, una interfaz de acceso a datos, y una base de datos. En la figura 2 se muestra un esquema de este tipo de sistemas, basado en la arquitectura de tres capas.



Fig. 1.2 Arquitectura de 3 capas de un sistema de información orientado a objetos

Capítulo 2

CAPÍTULO 2. DESCRIPCIÓN DEL SISTEMA AUTOMATIZADO CONTROL DE ASISTENCIA.

En el presente capítulo se realiza una descripción detallada del diseño de la base de datos utilizada en la aplicación, de las tablas que la conforman, la descripción de sus campos y las relaciones entre ellas. Se explica con detalles las tres capas que estructuran el software, Acceso a datos, Reglas del Negocio y Presentación mostrando de cada una de ellas sus métodos, y rasgos relevantes.

2.1 Selección de las herramientas

Para la realización del software se eligieron varias herramientas, se eligió el motor gestor de base de datos SQL Server 2005. El lenguaje de programación usado para confeccionar la interfaz visual y crear la funcionalidad del sistema fue el C#. Para la visualización del reporte se utilizó el *Crystal Report*, considerado el estándar para la creación de informes de Visual Studio 2005

2.2 Diseño de la base de datos

El diseño de una base de datos suele descomponerse en tres grandes fases (diseño conceptual, lógico y físico), lo que permite reducir la complejidad que entraña el diseño, a la vez que ayuda a alcanzar los dos principales objetivos que tienen las bases de datos:

- Ser una representación fidedigna del mundo real,
- Ser un servidor operacional y eficiente de los datos.

El diseño conceptual parte de la especificación de requerimientos, y produce como resultado el esquema conceptual de la base de datos. Un esquema conceptual es una descripción a alto nivel de la estructura de la base de datos, independientemente de la elección del equipamiento y del SGBD que se usen para la implementación de la base de datos.

El diseño lógico parte del esquema conceptual y genera el esquema lógico. Un esquema lógico es la descripción de la estructura de la base de datos que puede procesarse por un SGBD. Una vez elegido el modelo lógico, pueden existir un conjunto de esquemas lógicos equivalentes al mismo esquema conceptual. La meta del diseño lógico es producir el esquema lógico más eficiente con respecto a las operaciones de consulta y actualización.

El diseño físico toma como punto de partida el esquema lógico y como resultado produce el esquema físico. Un esquema físico es una descripción de la implementación de la base de datos en memoria secundaria; describe las estructuras de almacenamiento y los métodos de acceso para acceder a los datos de una manera eficiente. Por ello, el diseño físico se genera para un SGBD y un entorno físico determinado.

2.2.1 Diseño conceptual

El objetivo del diseño conceptual es obtener una buena representación de los recursos de información, independientemente del usuario o aplicaciones y fuera de consideraciones sobre eficiencia del ordenador. Consta de dos fases: análisis de los requisitos (¿qué representar?) y conceptualización (¿cómo representar?)^{[18][19]}.

Análisis de los requisitos: se desea establecer un modelo conceptual para la gestión del control de asistencia de los trabajadores a su centro laboral. De cada trabajador se desea conocer su nombre y apellidos, número de trabajador, área a la cual pertenece, ocupación, horario laboral, horas de entrada y de salida, horas de ausencia y causa. Al final de cada mes debe imprimirse un modelo con estos datos para cada trabajador y datos referidos al centro de trabajo al que pertenece: organismo, empresa, unidad, y a la persona responsable de la administración de este registro (ver Anexo II).

Conceptualización: esta etapa se apoya en el modelo Entidad-Relación, que se basa en una representación gráfica de una serie de entidades relacionadas entre sí. En la figura 3 se muestra el diagrama entidad relación para la base de datos Registro ^{[10][11]}.

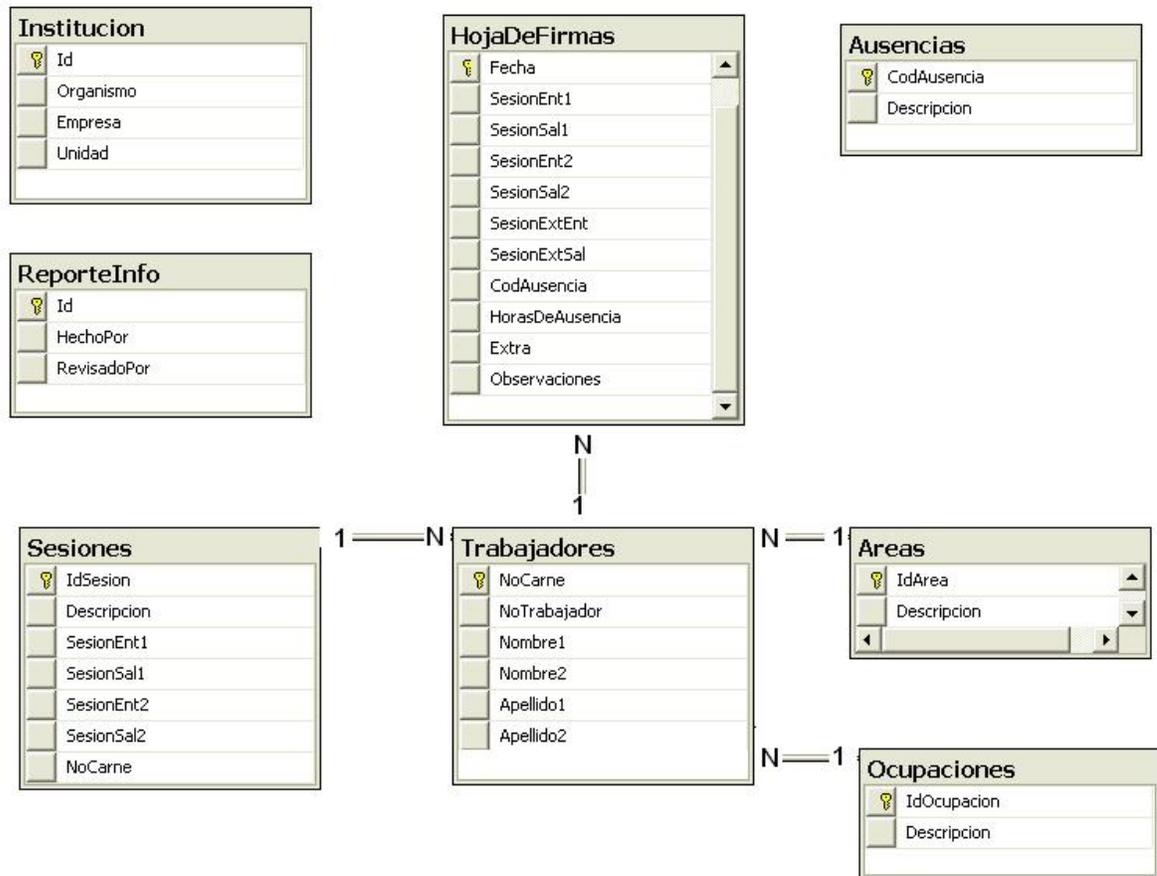


Fig. 2.1 Diagrama entidad relación para la base de datos Registro.

La cardinalidad expresa cuántas del conjunto de entidades de un extremo de la relación están relacionadas con cuántas entidades del conjunto del otro extremo. Pueden ser “uno a uno” expresado como 1 - 1, “uno a varios” o 1 - N y “varios a varios” que se representaría N - N ^[22].

Cardinalidad (Trabajador – Areas): N - 1

Cardinalidad (Trabajador – Ocupaciones): N - 1

Cardinalidad (HojaFirma – Trabajador): N - 1

Cardinalidad (Trabajador – Sesiones): N - 1

2.2.2 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 está muy ligado a la fase de diseño físico.

Para obtener el esquema lógico estándar a partir del esquema conceptual obtenido en la primera fase de diseño, se sigue un conjunto de reglas que permiten pasar del modelo entidad relación al esquema lógico:

- a. Cada entidad se transforma en una relación, con sus mismos atributos, incluyendo las claves.
- b. Cada relación N – M genera una tabla: las relaciones entre entidades con cardinalidad N – M generan una tabla, con los atributos clave de ambas entidades.
- c. Cada relación 1 – N importa las claves de la entidad con las que se relaciona: cada relación con cardinalidad 1 – N importa los atributos clave que contiene la entidad con cardinalidad N.
- d. Cada relación dependiente, importa la clave de otra entidad como clave.

En la figura 2.2 se muestra el esquema lógico resultante del esquema conceptual de la figura 2.1.

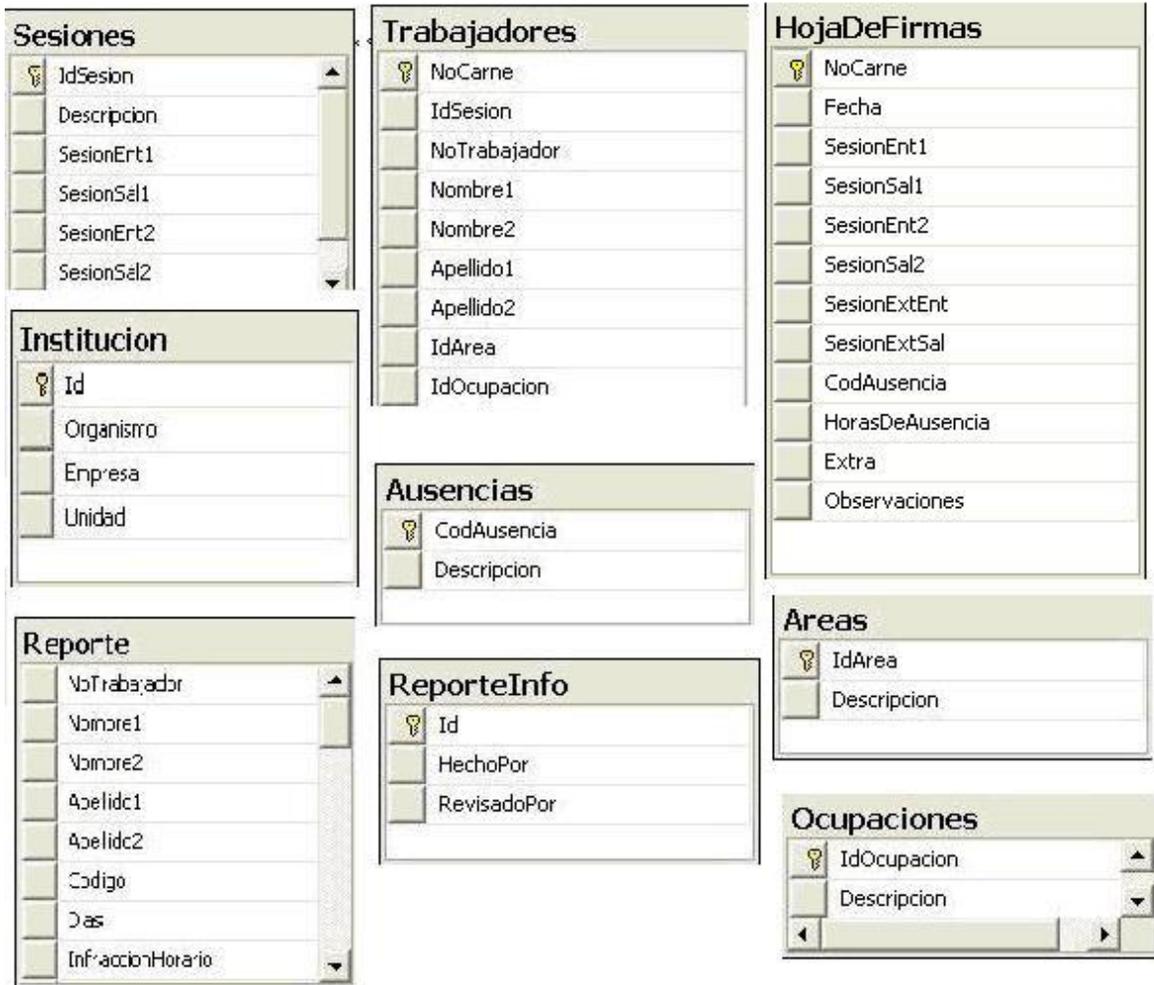


Figura 2.2 Esquema lógico de la base de datos Registro.

Descripción de las relaciones.

La relación *Trabajadores* contiene datos relativos a todos los trabajadores del centro. La llave primaria es el código de barras, NoCarne (bigint) que es un identificador único para cada trabajador del centro. Esta relación tiene interrelación con las relaciones,

La relación *HojaDeFirma* representa la vista del modelo de control de asistencia. Para cada trabajador (No_Carne) y una fecha (Fecha) guarda un registro de la entrada y salida de cada trabajador al centro, causa de ausencia (si existe), y cantidad de horas faltadas, entre otros. Esta relación se interrelaciona con la relación *Trabajador*.

La relación *Areas* contiene datos sobre el área de trabajo. La llave primaria es un número entero que identifica cada área que existe en el centro de trabajo.

La relación *Ocupación* contiene datos relativos a las diferentes ocupaciones existentes en el centro. La llave primaria es un número entero que identifica cada ocupación.

La relación *Sesiones* define los horarios de entrada y de salida de dos sesiones de trabajo y de una sesión extraordinaria (datetime). La llave primaria es un identificador de sesión, *IdSesion(int)* que identifica los turnos

La relación *Reporte* contiene los datos correspondiente al reporte de asistencia que se exportara de la aplicación.

La relación *Institución* contiene datos sobre el centro de trabajo. La llave primaria es un número entero que identifica una combinación de los campos *Organismo*, *Empresa*, y *Unidad*.

La relación *ReporteInfo* contiene datos sobre la administración del registro. La llave primaria es un número entero que identifica los responsables del registro de asistencia.

Dentro de nuestra base de datos encontramos una serie de procedimientos almacenados que garantizan la funcionalidad del sistema y la confiabilidad de las operaciones, éstos toman datos desde la aplicación y realizan operaciones que pueden devolver o no valores. Los procedimientos almacenados contemplados en la base de datos son:

- *TomarDatosInstitucion*: Selecciona los campos de la tabla *Institucion* devolviendo estos valores a la aplicación. Permitiendo que la interfaz de usuario en la capa de aplicación muestre al usuario la institución, empresa y unidad a la que pertenece.

- *RegistrarTrabajador*: Toma como entrada el número de carné de un trabajador, analiza la sesión de trabajo que le corresponde a este consultando la tabla *Sesiones*, luego en dependencia de la hora actual modifica el campo correspondiente (*SesionEnt1*, *SesionSal1*, *SesionEnt2*, *SesionSal2*) en la tabla *HojaDeFirmas*.
- *TomarDatosSesion*: Se le pasa como parámetro un número de carné de un determinado trabajador y devuelve los horarios de entrada y salida de las sesiones de trabajo del mismo.
- *TomarRegistrosDelDia*: Devuelve los usuarios que se han registrado en el día, con la primera corrida del programa no se mostrará nada ya que para esa fecha no hay usuarios registrados.
- *InsertarCodigo*: Este procedimiento se emplea para actualizar en la tabla *HojaDeFirmas* las causas de las usencias de los trabajadores. Toma los valores Clave, NoCarne y Fecha actualizando el campo correspondiente en la tabla *HojaDeFirmas*.

2.2.3 Diseño físico ^{[20][21]}.

El diseño físico es la expresión material del diseño lógico, incluye las cantidades y cualidades del hardware, software, y sistemas de aplicación.

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.

2.3 Modelo de tres capas aplicado al software

El Sistema de Control de Asistencia para trabajadores es un sistema típico de información que incluye una interfaz gráfica de usuario, una interfaz reglas del negocio, la interfaz de

acceso a datos, y una base de datos. En la siguiente figura se muestra el esquema de 3 capas para el software a realizar.

La aplicación consiste de una solución llamada *Controlador* con tres proyectos Bibliotecas de Clase (dll) que obedecen al nombre de las 3 capas del modelo multicapas, que se muestran en la siguiente figura.

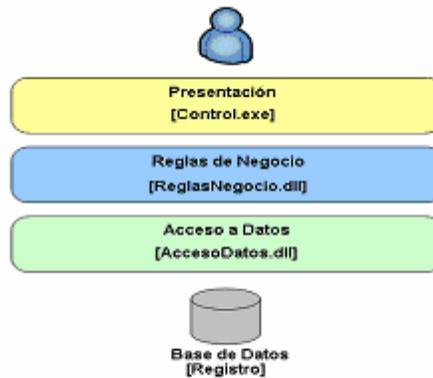


Fig. 2.3 Arquitectura de 3 capas del software.

2.3.1 La capa de acceso a datos

En esta capa se definen las funciones correspondientes a la conexión del software. Las funciones de esta capa se encuentran en el diagrama de clase que se muestra a continuación. Poner también los atributos, variables que se usan y explicarlas

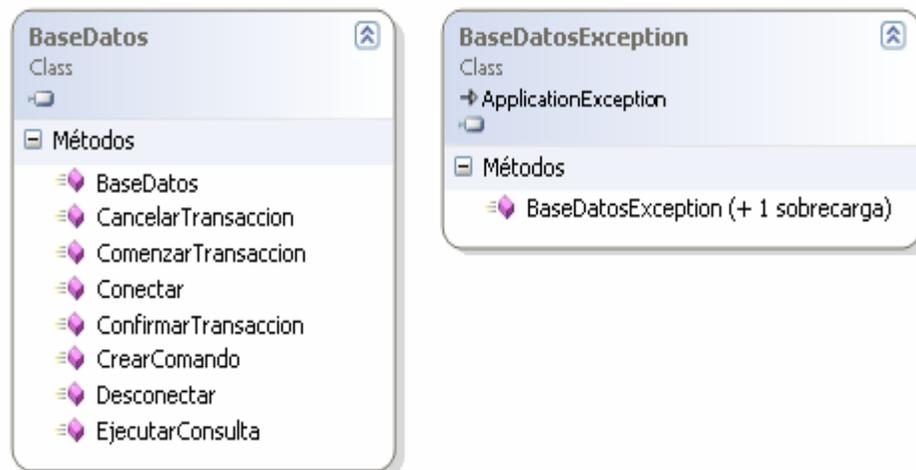


Fig. 2.4 Diagrama de clase BaseDatos.

Esta clase representa la base de datos del sistema, y se definen de qué forma la aplicación se va a comunicar con la base de datos, cada uno de los métodos posee una función específica, y muy concreta dentro de la clase.

- *Configurar*: se encarga de especificar los parámetros *Proveedor* y *Cadena_conexión*, 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* se encarga de crear un comando usando como origen una sentencia SQL, que es comprensible por el Sistema Gestor como una operación a realizar.
- *EjecutarConsulta* utiliza el comando creado por el método anterior y realiza la acción descrita por el comando.
- Luego que se terminan las operaciones se libera la conexión con la base de datos mediante el método *Desconectar*.

Si existe algún problema y la conexión no puede ser establecida se genera una excepción que es tratada en la clase *AccesoDatosException* mostrando un error al usuario.

2.3.2 La capa de negocio

En esta capa se definen las funciones *RegistrarTrabajador*, *TomarRegistrosdelDia*, *TomarDatosInstitución* y *InsertarCodigoAusencia* que son la que usando los datos a partir de la capa de aplicación y basándose en los métodos de la capa de datos permite ejecutar los procedimientos almacenados en la Base de Datos.

A continuación se muestra el diagrama de clase:

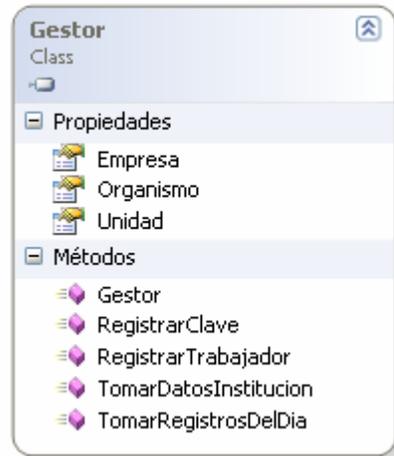


Fig. 2.5 Diagrama de clase *Gestor*.

- *RegistrarTrabajador*: es la encargada de enviarle a la base de datos el carnet del usuario que se está registrando en ese momento y que el procedimiento almacenado en la capa de datos se encargue de procesarlo e insertar en el campo registro la hora de llegada o salida.
- *InsertarCodigoAusencia*: se encarga de enviarle a la capa de datos el día, código de trabajador y clave de la ausencia de un trabajador para ser insertado en la tabla *HojaDeFirmas*.
- *TomarDatosInstitucion*: este método toma de la capa de datos los datos de la institución a la que pertenece la aplicación, esta puede variar en dependencia de en qué centro se instale la aplicación.
- *TomarRegistrosDelDia*: toma los valores de los trabajadores que se han registrado a lo largo del día.

En cada uno de estos métodos se usa una excepción que se encarga de mostrar un error que esté en correspondencia con el error que provocó la excepción. Evitando que el usuario se enfrente a un error del sistema, que es muy probable que no comprenda. En su lugar se mostrará un error con un mensaje previsto de antemano por el programador, que a su vez facilita su reparación ya que se sabe el punto exacto donde se produce el fallo.

2.3.3 La capa de presentación o aplicación.

En la capa de presentación es donde se encuentra la interfaz de usuario para facilitar la interacción de éste 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 ésta es la encargada de tramitar con la capa de datos las acciones realizadas por el usuario sobre la base de datos. A través de la interfaz de usuario será posible una serie de acciones como el registro de los trabajadores, la inserción de códigos de ausencia para días faltados, la obtención del reporte de ausencia mensual (Modelo RH-3, ver anexo II).

La interfaz del reporte se genera usando el componente *CrystalReportViewer* que obtiene los datos almacenados en la tabla *Reporte* de nuestra base de datos. Dentro de la interfaz del reporte se pueden hacer además otras acciones como imprimir, hacer búsquedas de datos, exportar el informe a distintos formatos como pueden ser doc, pdf, xls, rpt etc.

Un papel importante dentro de la interfaz de usuario lo juega el componente que constantemente se actualiza, cada vez que se registra un trabajador, mostrando los campos de entrada y salida. Se usó para esta tarea el componente *DataGridView* que toma como origen de datos un filtrado de la tabla *HojaDeFirmas* de nuestra base de datos.

A continuación se muestra un diagrama de la clase *InterfazPrincipal.cs* donde se puede ver los métodos y variables que forman la funcionalidad de la clase, así como algunas de los componentes visuales que componen la interfaz visual.

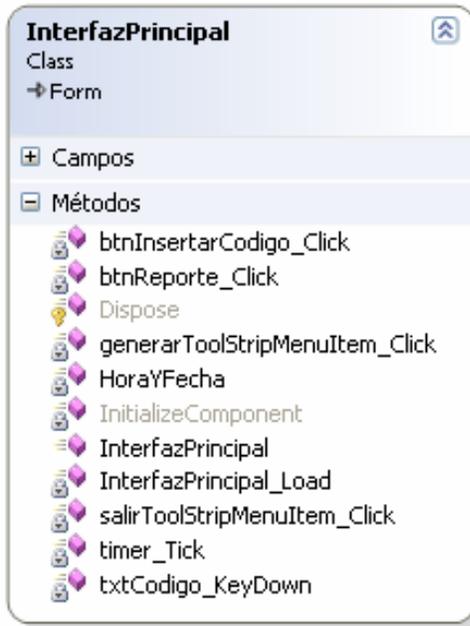


Figura 2.6 Diagrama de clase de la clase *InterfazPrincipal*.

En la figura anterior se pueden ver algunos de los métodos que se usan y que forman parte de la funcionalidad de la interfaz de usuario a continuación una explicación de éstos.

- *txtCodigo_KeyDown* Responde a la entrada de un código de barras por el usuario ya sea insertado por el escáner, o directamente por el operador de manera manual.
- *HoraYFecha*: es el método que se encarga de refrescar en cada momento la fecha y hora del sistema, mostrando el día de la semana, la fecha actual con el formato dd/MM/yyyy h:mm:ss
- *Reporte_Click*: es el encargado de cargar el Visor de Informes con los datos resumen de las ausencias mensuales que se encuentran en la base de datos.
- *InsertarCodigo_Click*: es quien ejecuta la acción de mostrar una nueva interfaz para insertar el código de ausencia de un trabajador “x” en una determinada fecha.

La figura siguiente muestra el diagrama de la clase Visor de Informes que es la encargada de mostrar los datos con el resumen de las ausencias mensuales.

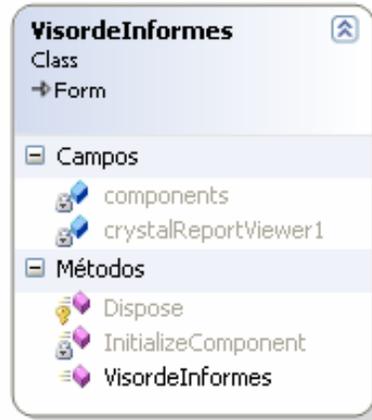


Figura 2.7 Diagrama de clase *VisordeInformes*.

Crystal Reports es el estándar de creación de informes de Visual Studio 2005. Se incluye en todas las copias de Visual Studio 2005 y se integra directamente en el entorno de desarrollo.

Crystal Reports para Visual Studio 2005 puede utilizarse para interactuar con el informe, usando uno de los cuatro modelos de objetos posibles:

- `CrystalReportViewer`, el modelo de objetos más sencillo.
- `ReportDocument`, el modelo de objetos más completo.
- `ReportClientDocument`, el modelo de objetos más completo.
- `InfoObject`, un modelo de objetos muy eficaz para la programación y configuración de informes en el marco de `Crystal Reports Server` o `BusinessObjects Enterprise`.

En nuestro caso se usó el modelo de objeto *CrystalReportViewer*, este objeto tiene como origen de datos una tabla de nuestra base de datos que posee el resumen de asistencia que se va a mostrar, y haciendo coincidir los campos del *CrystalReportViewer* con los campos de tabla en la base de datos se muestra un informe con muchas funcionalidades y comodidades para el usuario.

Capítulo 3



CAPÍTULO 3. MANUAL DE USUARIO.

Para el adecuado uso del software es necesaria una breve introducción, para que el usuario se familiarice con el mismo y así lograr un mejor desempeño del mismo. En este apartado se explicarán las funciones básicas del software así como su modo de manejo, con lo cual se pretende un ahorro de tiempo al procesar los datos.

Al iniciar el programa se muestra una tabla con los trabajadores con entradas o salidas ese día, que estará vacía si es la primera vez en el día que se inicia el software. Estos datos están almacenados en una base de datos que puede estar en la propia computadora o ser consultados de manera remota en otro servidor.

Cada trabajador posee, para el caso de la Universidad Central, un carné con un código de barras que lo identifica, este código es captado y procesado por un lector de códigos de barras e insertado en la base de datos en el campo marcado con un “1” en la siguiente figura, además, se pueden entrar los datos de manera manual por teclado pasando los números del carné de identidad correspondientes.

La siguiente figura muestra una corrida del programa cuando se inicia por primera vez en el día.

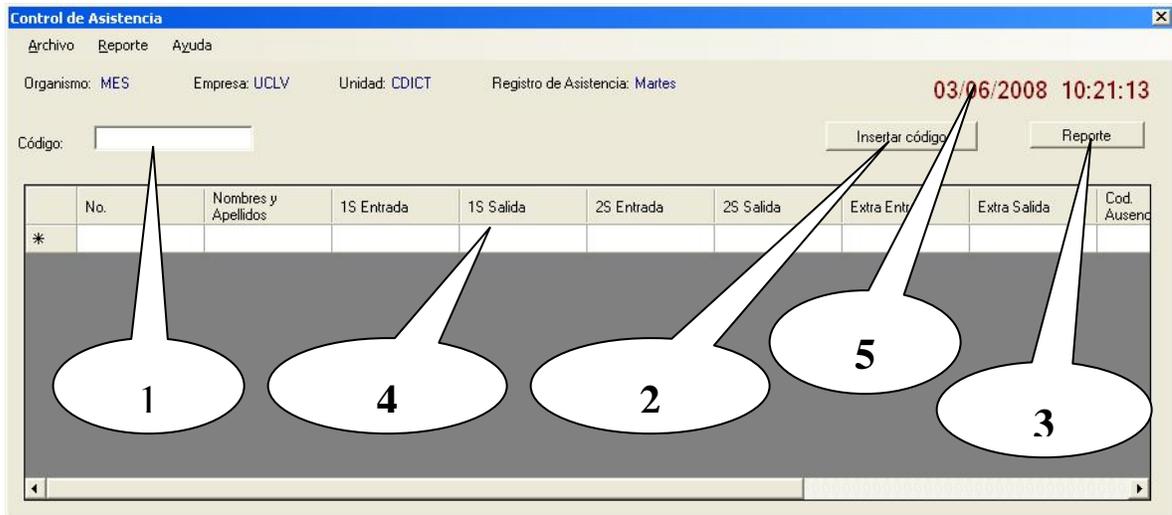


Figura 3.1 Primera ejecución del día.

El campo señalado con “1” es quien recibe los datos desde el código de barras o la inserción manual de los números correspondientes al carné del trabajador.

Se muestra el botón “2”, este se utiliza para insertar la clave de ausencia por la que un trabajador “x” se ausenta al centro laboral, desplegando una nueva ventana como la que se muestra a continuación, este código es del conocimiento de los encargados de llevar el control de la asistencia en las empresas. Con la entrada del carné de identidad del usuario, fecha de la ausencia y clave se actualizará la base de datos.

El botón señalado con un “3” es el encargado de generar el reporte con el resumen de las ausencias de cada trabajador y las infracciones de horario, otra forma de acceder a esta acción es desplegar el menú reporte y marcar la opción generar. El componente señalado con un “4” es un *DataGridView* que se llenará a medida que vayan llegando los trabajadores, actualizándose constantemente.

Esta interfaz muestra el organismo al que pertenece dicha aplicación, este dato se puede modificar en la base de datos de manera que pueda ser utilizado en cualquier empresa o centro de estudios del país. A la derecha se muestra el día de la semana, así como la fecha y hora que posee el sistema (campo marcado con “5”). Es muy importante que la hora del sistema sea correcta porque la aplicación hace comparaciones de la hora actual para llenar

los registros de los trabajadores de entrada o salida de manera que un error en la hora del sistema no puede permitirse para el correcto desempeño del software



Figura 3.2 Ventana que permite insertar el código de ausencia de un trabajador.

A medida que se van llenando los datos de los trabajadores la tabla va tomando la forma que se muestra a continuación.

The image shows the "Control de Asistencia" application window. At the top, there is a menu bar with "Archivo", "Reporte", and "Ayuda". Below the menu bar, the status information includes "Organismo: MES", "Empresa: UCLV", "Unidad: CDICT", and "Registro de Asistencia: Viernes". The date and time "13/06/2008 10:28:10" are displayed in red. There are input fields for "Código:", "No. Trab.", "Dia" (with a red label "Mes/Día/Año"), and "Clave", along with "Insertar" and "Reporte" buttons.

No.	Nombres y Apellidos	1S Entrada	1S Salida	2S Entrada	2S Salida	Extra Entrada	Extra Salida	Cod. Ausenc
777	HENRY HERRE...	13/06/2008 7:02	13/06/2008 11:00	13/06/2008 13:04	13/06/2008 17:00			
789	Adrian Doss Sant...	13/06/2008 7:00	13/06/2008 11:00	13/06/2008 13:05	13/06/2008 17:00			
333	Yoel Alvarez Rojas	13/06/2008 7:00	11/06/2008 11:00	11/06/2008 13:00	11/06/2008 17:00			
444	Juan Hernandez...	13/06/2008 7:04	11/06/2008 11:00	11/06/2008 13:00	11/06/2008 17:00			
111	Aleocha Garcia ...	13/06/2008 7:00	11/06/2008 11:00	11/06/2008 13:00	11/06/2008 17:00			
516	DAIRO BLANCO ...	13/06/2008 7:00	11/06/2008 11:00					
555	Yury Carvajal Mer...	13/06/2008 7:00	11/06/2008 10:23	11/06/2008 13:00	11/06/2008 16:45			
*								

Figura 3.3 Ejecución desprograma con datos.

La generación del reporte muestra una interfaz como la que aparece a continuación.

UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS DIRECCION DE RECURSOS HUMANOS		REPORTE DE AUSENCIAS Y TRABAJO REALIZADO		MODELO RH-3																		
		Unidad CDICT																				
No. Trab.	APELLIDOS Y NOMBRES	DISTRIBUCION DE TIEMPO														RESUMEN TIEMPO NO TRABAJADO				INFRACCION DE HORARIO		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	COD	COD	COD	COD	CANT. MTOS.	
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	DIAS	DIAS	DIAS	DIAS	
555	Yury Carvajal Merino				x			x								x		03				1,00
111	Denis Ramoncito Pacheco	x		x			x			x								01				5,00
333	Yoel Alvarez Rojas																	0				
222	Enrique San Gil Hernandez															x		04				12,30
516	Dairo Blanco Guevara					x		x										05				
777	Henry Herrera Ezcurdia																	2				44,10
444	Juan Hernandez Lucho				x	x	x	x	x	x	x							04				
789	Adrian Dos Santos																	7				
																		0				

HECHO POR	REVISADO POR	RECURSOS HUMANOS		FECHA	
		REVISADO	APROBADO	MES	Año

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Figura 3.4 Muestra la interfaz del reporte.

Esta interfaz nos da la posibilidad de realizar operaciones como búsquedas dentro del reporte de nombres o datos, impresión de manera directa, así como es capaz de exportar los datos del reporte a distintos formatos como rpt, doc, pdf, xls que quedaría a comodidad del usuario, como se muestra continuación.

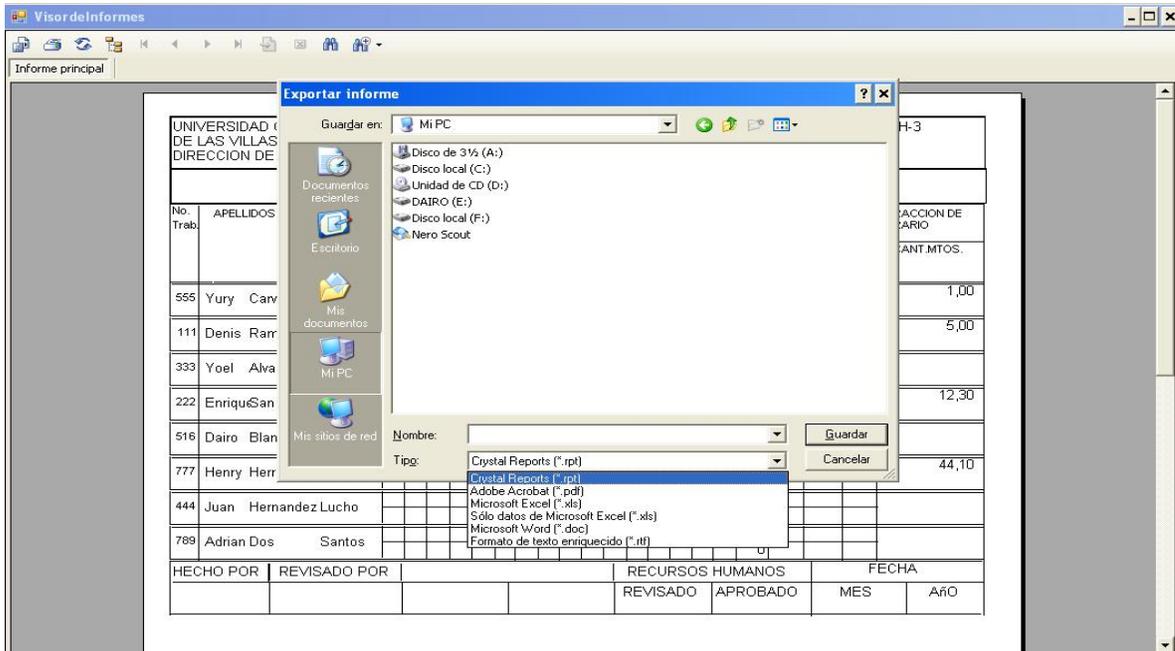


Figura 3.5 Muestra la exportación de datos del reporte.

En la siguiente figura se observa un ejemplo de una búsqueda dentro del reporte que puede ser muy útil para el usuario.

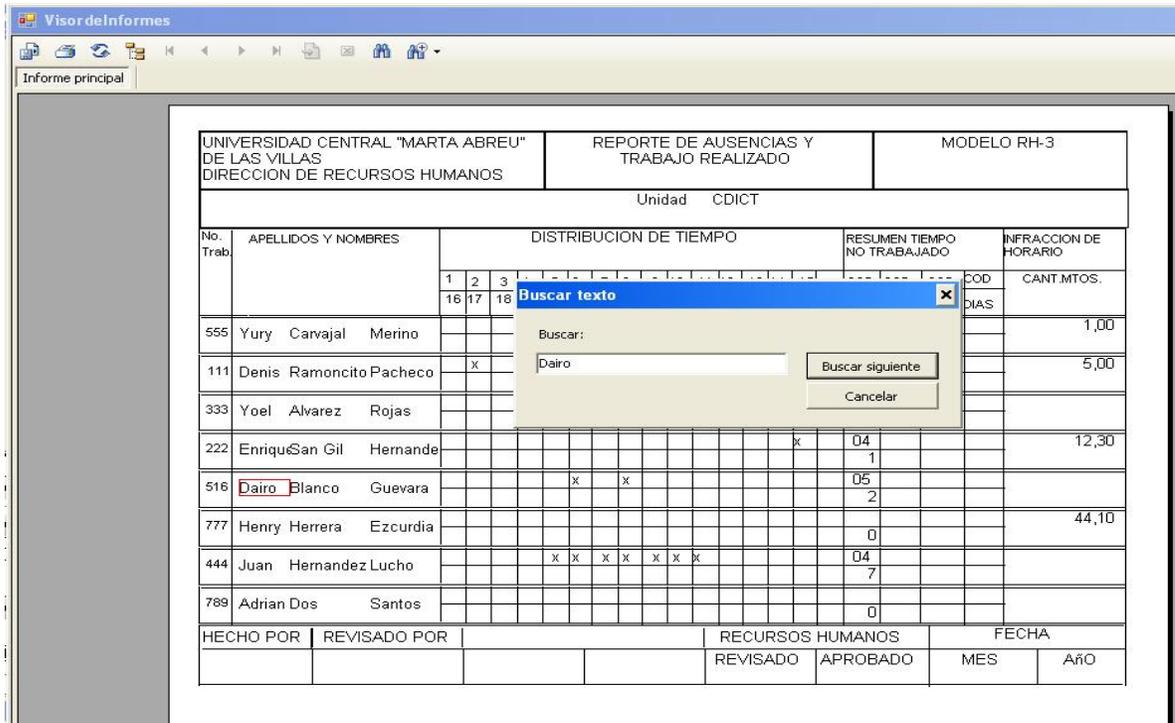


Figura 3.6 Búsquedas de textos dentro del reporte.

Conclusiones

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Con este trabajo se logra como resultado la obtención de un software para el control de la asistencia de los trabajadores del CDICT utilizando el C# 2.0 y el SQL Server 2005.

El software es un ejecutable para Windows, con una interfaz gráfica que lo hace cómodo para su explotación y con una estructura que facilita su extensibilidad.

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

- La realización de un software de este tipo mejora la calidad del proceso de identificación de los usuarios en el centro, lo hace 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 y actualmente en desarrollo, explotados en prácticamente todos los países en medianas y grandes empresas.
- La utilización de software libre como los lenguajes que pertenecen a la plataforma .NET como el CSHARP son mundialmente utilizados para la realización de aplicaciones, así como lo es el SQL Server como gestor de bases de datos.

Recomendaciones.

Recomendaciones

Se recomienda la generalización del software primeramente en la Universidad Central, y que en un futuro puede abarcar a todo el país. Se puede recomendar también que se le agregue mayores funcionalidades en cuanto a las operaciones a realizar. Enlazar varios sistemas como éste localizados en distintos lugares y que envíen los datos de los reportes automáticamente, por ejemplo, al personal de recursos humanos vía correo electrónico u otra vía con el fin de ser procesados.

Referencias Bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

[1] Anónimo: ABC del código de barras. Artículo publicado en Internet.

URL: www.arzp.com/bar=code/lectores.html

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

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

[3] Sofía, Marcelo: Guía de referencia EAN/UCC-14. Artículo de la firma EAN Argentina-código.

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

URL: www.monografias.com

[5] Anónimo: Uso de Códigos de Barras para Principiantes. Artículo de la firma Symbol Technologies. Inc.

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

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

[7] Manual de usuario EAN.UCC.

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

[8] Anónimo: CODIGO UCC/EAN-128. Artículo de EAN-Venezuela.

URL: www.eanve.org

[9] Azalea Software, Inc. 2007, Barcode Software & Bar Code Fonts.

URL: www.azalea.com

[10] Moratalla, J. (2001). *Bases de Datos con SQL Server 2000. Transact SQL*

[11] Raghu Ramakrishnan, J. G. (2000). *Database Management Systems* (Second Edition ed.).

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

[13] Archer, T. (2001). *A Fondo C#*.

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

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

[16] Larman, C. (1999). *UML y Patrones*: Pearson.

[17] Lewis, G. (1994). *What is Software Engineering?*. DataPro (4015). pp. 1-10.

[18] Cota, A. (1994). *Ingeniería de Software*. Soluciones Avanzadas. pp. 5-13.

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

[20] hdolder.com srl 2008, M&P Requerimientos de Hard y Soft

URL: http://www.hdolder.com/C_948.htm

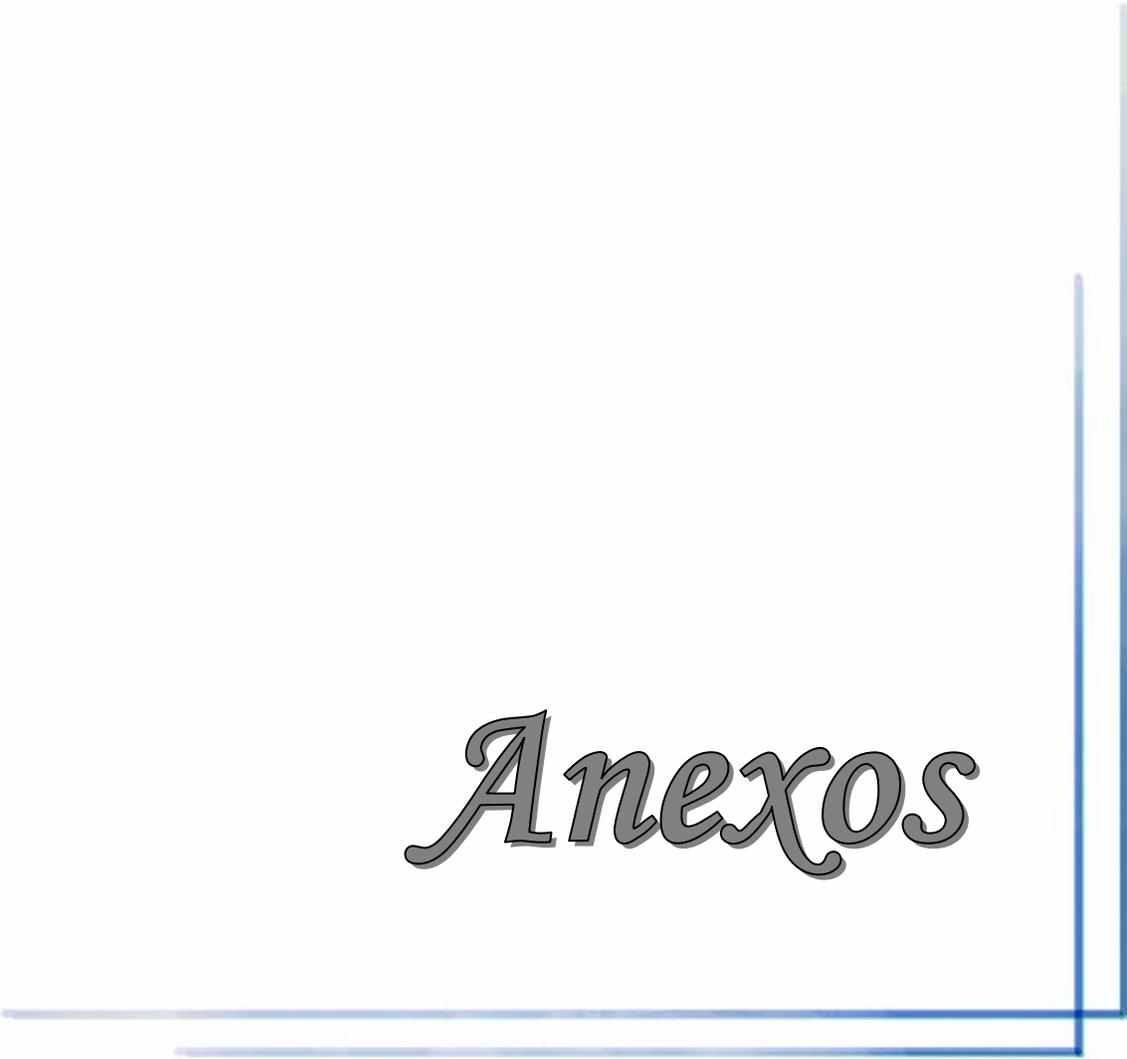
[21] Microsoft Corporation 2008, Requisitos del sistema SQL Server 2005

URL: <http://www.microsoft.com/spain/sql/productinfo/sysreqs/default.mspx>

[22] Herrera, Álvaro 2004-10-04, Cardinalidad

URL: <http://www.alvherre.cl/pgsql/modBasico/node6.html>

[23] Ferguson, Jeff.(2003). *La Biblia C#*



Anexos

