



Universidad de las Ciencias Informáticas
Facultad 2

**SISTEMA CONTROL DE ACCESO A COMEDORES
EN LA
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores:

**Adisley Flores Coronado
Imirys Rovira Prieto**

Tutor:

Ing. Manuel Alejandro Gil Martín

Ciudad de La Habana, Cuba

Junio, 2007



DECLARACIÓN DE AUTORÍA

Declaración de Autoría

Declaro que somos los únicos autores de este trabajo y autorizamos a la Dirección de Informatización de la Universidad de las Ciencias Informáticas; así como a dicho centro para que se use este trabajo en lo que se estime conveniente.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autores: Adisley Flores Coronado, Imirys Rovira Prieto

Tutor: Manuel Alejandro Gil Martín

OPINIÓN DE LOS USUARIOS

Opinión de los usuarios

El Trabajo de Diploma, titulado “Sistema Control de Acceso a Comedores en la Universidad de las Ciencias Informáticas”, fue realizado en la propia Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente ____

Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a ____.

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR

Opinión del Tutor

Título: Sistema Control de Acceso a Comedores en la UCI

Autores: Adisley Flores Coronado, Imirys Rovira Prieto

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

Tutor: Manuel Alejandro Gil Martín

Firma

Fecha

AGRADECIMIENTOS

Agradecimientos de Imirys

Agradecer a mis papás por su apoyo, a Imi, a mi familia toda...

A mis amigos, que los voy a extrañar mucho, por tolerarme y quererme.

A mis profes, los de verdad...

A mis alumnos, por hacerme estudiar.

A Chony, mi tutor, por toda su ayuda.

A Marpa, al Bruji, a Chuchi, a Ady, a Made y a Jose, a todos mis compañeros de campaña.

Los quiero a todos...

Imy

AGRADECIMIENTOS

Agradecimientos de Adisley

Quisiera agradecer a mi familia, por todo lo que han podido aportar en mi formación, especialmente a mi mamá, a mi hermana Amairelys y a mi cuñado Yoel.

Un profundo agradecimiento a mi tutor por sus consejos, ideas y guía.

A todos mis amigos que me apoyaron en cada momento para que pudiera seguir adelante y que hicieron que estos años fueran inolvidables, especialmente a Maybel, Leydis, Marisleydis, Aislein, Imirys, Maydolis, Yudier y Alain.

A todos mis compañeros, por haberme ayudado y con los que he compartido estos cinco años.

A los profesores y a la Universidad por haber hecho posible formarme como profesional.

En fin, para no olvidar a nadie, a todos aquellos que de una forma u otra me han brindado su apoyo, a todos...

Muchas Gracias.

Ady

DEDICATORIA

A mis seres más queridos...

Imy

A mi madre y a mi hermana...

Ady

En la actualidad es considerado un problema el hecho de gestionar el acceso a los comedores en la Universidad de las Ciencias Informáticas (UCI) de tantos comensales en las distintas sesiones de alimentación. Este es un proceso de suma importancia en el contexto del flujo diario de actividades que conforman el funcionamiento interno de la Universidad. El sistema que automatiza estos procesos hoy en día funciona desde hace algunos años, pero debido a que el número de personas que conviven y/o trabajan en la Universidad se ha incrementado en estos últimos tiempos, para que el mismo continuara prestando servicios fue preciso ir encontrando soluciones temporales que con el tiempo pasaron a ser permanentes, comprometiendo el buen funcionamiento de la aplicación. Por esta razón se llegó a la conclusión de que el sistema de acceso a comedores actual de la UCI no permite una adecuada gestión para las situaciones que se presentan.

Para resolver este problema se realizó una propuesta de solución, así surge CONTACC, el sistema de control de acceso a comedores en la UCI. Este sistema tiene como objetivo fundamental gestionar los accesos a comedores, ello consiste en: registrar accesos, comprobando que dicha persona existe, tiene autorización para pasar por esa puerta y no ha accedido con anterioridad, sincronizar accesos o actualizar cada cierto tiempo la información registrada en las bases de datos locales con las bases de datos centrales a través del Servicio Web; y brindar un servicio de reportes donde se muestre al usuario la cantidad de personas (por tipo) que han tenido acceso hasta el momento.

CONTACC se desarrolló utilizando el Gestor de Bases de Datos SQLite para las bases de datos locales y un Servidor Web con el que intercambia información, el lenguaje de programación utilizado fue C#, y la plataforma .Net. Se empleó RUP como metodología de desarrollo y UML como lenguaje de modelado. Además utiliza la tecnología de código de barras para identificar a las personas.

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 INTRODUCCIÓN.	5
1.2 ESTADO DEL ARTE NACIONAL E INTERNACIONAL.	5
1.2.1 <i>Estado del arte nacional</i>	5
1.2.2 <i>Estado del arte internacional</i>	5
1.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	8
1.3.1 <i>Fundamentación de las tecnologías</i>	8
1.3.2 <i>Tecnologías de hardware utilizadas para el control del acceso</i>	9
1.3.3 <i>Tecnologías de software utilizadas</i>	14
1.4 LA PROPUESTA.	25
1.5 CONCLUSIONES.	26
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	27
2.1 INTRODUCCIÓN.	27
2.2 OBJETO DE ESTUDIO.	27
2.2.1 <i>Problema y situación problemática</i>	27
2.2.2 <i>Objeto de automatización</i>	28
2.2.3 <i>Información que se maneja</i>	29
2.3 PROPUESTA DE SISTEMA.	29
2.4 MODELO DE NEGOCIO.	30
2.4.1 <i>Reglas del negocio</i>	30
2.4.2 <i>Descripción de actores y trabajadores</i>	31
2.4.3 <i>Diagrama de Casos de Uso del Negocio</i>	31
2.4.4 <i>Descripción de casos de uso del negocio. Anexo 1</i>	32
2.4.5 <i>Diagramas de Actividad. Anexo 1</i>	32
2.4.6 <i>Modelo de Objetos. Anexo 1</i>	32
2.4.7 <i>Listado de los requerimientos funcionales</i>	32
2.5 MODELO DE CASOS DE USO DEL SISTEMA.	36
2.5.1 <i>Definición de los actores del sistema a automatizar</i>	36

2.5.2 Listado de casos de uso del sistema.	36
2.5.3 Diagrama de casos de uso del sistema a automatizar.	39
2.5.4 Descripción de los casos de uso del sistema. Anexo2	40
2.6 CONCLUSIONES.	40
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA	41
3.1 INTRODUCCIÓN.	41
3.2 ANÁLISIS.....	41
3.2.1 Modelo de clases de análisis.	41
3.3 DISEÑO.	45
3.3.1 Diagramas de interacción. Anexo 3.....	45
3.3.2 Diagrama de clases del diseño.	45
3.4 DISEÑO DE LA BASE DE DATOS.....	54
3.4.1 Modelo lógico de datos (diagrama de clases persistentes).	54
3.4.2 Modelo físico de datos (modelo de datos).	55
3.4.3 Descripción de las tablas. Anexo 4.....	55
3.5 ARQUITECTURA DE CAPAS.	55
3.6 DEFINICIONES DE DISEÑO QUE SE APLIQUEN.	57
3.6.1 Interfaz de usuario.	58
3.7 TRATAMIENTO DE ERRORES.	61
3.8 CONCLUSIONES.	61
CAPÍTULO 4. IMPLEMENTACIÓN.....	62
4.1 INTRODUCCIÓN.	62
4.2 DIAGRAMA DE DESPLIEGUE.....	62
4.2.1 Representación del diagrama de despliegue.....	63
4.3 Diagrama de componentes.....	63
4.3.1 Representación del diagrama de componentes.....	64
4.4 CONCLUSIONES.	64
CAPÍTULO 5. ESTUDIO DE FACTIBILIDAD	65
5.1 INTRODUCCIÓN.	65
5.2 PLANIFICACIÓN BASADA EN CASOS DE USO.	65

5.3 BENEFICIOS TANGIBLES E INTANGIBLES.....	71
5.4 ANÁLISIS DE COSTOS Y BENEFICIOS.....	72
5.5 CONCLUSIONES.....	72
CONCLUSIONES GENERALES.....	73
RECOMENDACIONES.....	74
REFERENCIAS BIBLIOGRÁFICAS	75
BIBLIOGRAFÍA	76
ANEXOS.....	79
GLOSARIO DE TÉRMINOS.....	96

ÍNDICE DE TABLAS

TABLA 1. TECNOLOGÍAS BIOMÉTRICAS.....	10
TABLA 2. ACTORES DEL NEGOCIO.....	31
TABLA 3. TRABAJADORES DEL NEGOCIO.....	31
TABLA 4. ACTORES DEL SISTEMA.....	36
TABLA 5. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “REGISTRAR_ACCESO”.....	36
TABLA 6. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “SINCRONIZAR_PERSONA”.....	37
TABLA 7. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “SINCRONIZAR_ACCESO”.....	37
TABLA 8. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “MOSTRAR_DATOS”.....	37
TABLA 9. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “MOSTRAR_REPORTES”.....	37
TABLA 10. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “AUTENTICAR”.....	38
TABLA 11. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “MOSTRAR_DATOS_CONFIGURACIÓN”.....	38
TABLA 12. DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA “ACTUALIZAR_DATOS_CONFIGURACIÓN”.....	38
TABLA 13. FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR.....	66
TABLA 14. FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR.....	66
TABLA 15. FACTOR DE COMPLEJIDAD TÉCNICA.....	68
TABLA 16. FACTOR DE AMBIENTE.....	69
TABLA 17. ESFUERZO DEL PROYECTO.....	70
TABLA 18. DIAGRAMA DE ACTIVIDAD “ACCEDER AL COMEDOR”.....	79
TABLA 19. “REGISTRAR_ACCESO”.....	82
TABLA 20. “SINCRONIZAR_PERSONA”.....	83
TABLA 21. “SINCRONIZAR_ACCESO”.....	83
TABLA 22. “MOSTRAR_DATOS”.....	84
TABLA 23. “MOSTRAR_REPORTES”.....	84
TABLA 24. “AUTENTICAR”.....	85
TABLA 25. “MOSTRAR_DATOS_CONFIGURACIÓN”.....	85
TABLA 26. “ACTUALIZAR_DATOS_CONFIGURACIÓN”.....	86
TABLA 27. “PERSONA”.....	94
TABLA 28. “ACCESOSPERSONA”.....	94
TABLA 29. “USUARIO”.....	95

ÍNDICE DE TABLAS

TABLA 30. "CONFIGURACIÓN"	95
TABLA 31. "PUERTA"	95
TABLA 32. "TOTALES"	95

ÍNDICE DE FIGURAS

FIGURA 1. RELOJ CHEQUEADOR UCONTROL 6000 ETHERNET.....	6
FIGURA 2. RELOJ CHEQUEADOR HANDPUNCH O TERMINAL LECTORA DE MANO.....	7
FIGURA 3. RELOJ CHEQUEADOR BIOCONTROL 1000/A11.....	7
FIGURA 4. CÓDIGO 39.....	13
FIGURA 5. LECTOR PORTÁTIL DE CÓDIGO DE BARRAS (VOYAGER DE METROLOGIC).....	14
FIGURA 6. DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	32
FIGURA 7. DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	39
FIGURA 8. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “AUTENTICAR”.....	41
FIGURA 9. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “MOSTRAR_DATOS_CONFIGURACIÓN”.....	42
FIGURA 10. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “ACTUALIZAR_DATOS_CONFIGURACIÓN”.....	42
FIGURA 11. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “MOSTRAR_DATOS”.....	43
FIGURA 12. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “MOSTRAR_REPORTE”.....	43
FIGURA 13. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “REGISTRAR_ACCESO”.....	44
FIGURA 14. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “SINCRONIZAR_ACCESO_PERSONA”.....	44
FIGURA 15. DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO “SINCRONIZAR_PERSONA”.....	45
FIGURA 16. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “AUTENTICAR”.....	46
FIGURA 17. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “MOSTRAR_DATOS_CONFIGURACIÓN”.....	47
FIGURA 18. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “ACTUALIZAR_DATOS_CONFIGURACIÓN”.....	48
FIGURA 19. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “MOSTRAR_DATOS”.....	49
FIGURA 20. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “MOSTRAR_REPORTE”.....	50
FIGURA 21. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “REGISTRAR_ACCESO”.....	51
FIGURA 22. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “SINCRONIZAR_ACCESO_PERSONA”.....	52
FIGURA 23. DIAGRAMA DE CLASES DE DISEÑO DEL CASO DE USO “SINCRONIZAR_PERSONA”.....	53
FIGURA 24. DIAGRAMA DE CLASES PERSISTENTES.....	54
FIGURA 25. MODELO DE DATOS.....	55
FIGURA 26. ARQUITECTURA EN CAPAS.....	57
FIGURA 27. INTERFAZ DE USUARIO.....	58
FIGURA 28. PERSONAS.....	59
FIGURA 29. CONFIGURACIÓN.....	60

ÍNDICE DE FIGURAS

FIGURA 30. DIAGRAMA DE DESPLIEGUE.....	63
FIGURA 31. DIAGRAMA DE COMPONENTES.....	64
FIGURA 32. DIAGRAMA DE ACTIVIDAD “ACCEDER AL COMEDOR”.....	80
FIGURA 33. MODELO DE OBJETO.....	81
FIGURA 34. DIAGRAMA DE SECUENCIA DEL CASO DE USO “AUTENTICAR”.....	87
FIGURA 35. DIAGRAMA DE SECUENCIA DEL CASO DE USO “MOSTRAR_DATOS_CONFIGURACIÓN”.....	87
FIGURA 36. DIAGRAMA DE SECUENCIA DEL CASO DE USO “ACTUALIZAR_DATOS_CONFIGURACIÓN”.....	88
FIGURA 37. DIAGRAMA DE SECUENCIA DEL CASO DE USO “MOSTRAR_DATOS”.....	89
FIGURA 38. DIAGRAMA DE SECUENCIA DEL CASO DE USO “MOSTRAR_REPORTE”.....	90
FIGURA 39. DIAGRAMA DE SECUENCIA DEL CASO DE USO “REGISTRAR_ACCESOS”.....	91
FIGURA 40. DIAGRAMA DE SECUENCIA DEL CASO DE USO “SINCRONIZAR_ACCESO_PERSONA”.....	92
FIGURA 41. DIAGRAMA DE SECUENCIA DEL CASO DE USO “SINCRONIZAR_PERSONA”.....	93

INTRODUCCIÓN

A lo largo de la historia, ha existido la necesidad de controlar el acceso a ciertas áreas y lugares, esta necesidad, motivada inicialmente por el propio sentimiento de pertenencia, resulta el motor impulsor para que comenzaran a utilizarse mecanismos de control de accesos.

Hoy en día, tecnológicamente se ha evolucionado, y en este sentido se han desarrollado y automatizado sistemas que se encargan de controlar el acceso sobre áreas e información que se desea proteger.

Un gran número de personas tienen acceso a los comedores en los diferentes horarios de desayuno, almuerzo y comida en la Universidad de las Ciencias Informáticas (UCI), donde es considerado un problema el hecho de gestionar el acceso a los mismos de tantos comensales.

En el centro ya existe un sistema que automatiza estos accesos, pero debido a que el número de personas que conviven y/o trabajan en la Universidad se ha incrementado en estos años, producto de los nuevos ingresos de estudiantes y de las contrataciones de servicios de trabajo, entre otras razones, se ha ido comprometiendo el buen funcionamiento del mismo, pues para que el sistema continuara prestando servicios fue preciso ir encontrando soluciones temporales que con el tiempo pasaron a ser permanentes. Todo ello conlleva al **problema**: El sistema de acceso a comedores actual de la UCI no permite una adecuada gestión para las situaciones que se presentan hoy en día.

La Universidad requiere de un nuevo sistema que elimine las deficiencias de esta aplicación que hoy funciona, pues esta gestión tiene que ser más dinámica, garantizando la seguridad de la información y no tener que hacer grandes cambios futuros; definiéndose la **situación problemática** como: El actual sistema de acceso a comedores de la UCI no puede ajustarse a las nuevas condiciones de la Universidad, puesto que está muy ligado a las reglas de negocios sobre las que se desarrolló. Además es una aplicación Web con réplicas de los servidores de bases de datos de la Universidad en cada uno de los complejos de comedores, por lo que funciona sin estar conectado directamente a los servidores centrales, esto trae consigo que haya más información de la necesaria en cada uno de ellos. A esto se suma que la aplicación

presenta demoras en mostrar los datos de las personas que acceden y las actualizaciones de las réplicas sólo se realizan en determinados momentos, por lo que no siempre se tiene la información de última hora.

Así, surge el Sistema de Control de Acceso a Comedores en la UCI. Con el uso de esta aplicación se mejorará el proceso de gestión de acceso a comedores ya que las distribuciones de las personas a acceder por cada puerta serán proporcionadas por un Servidor Web y registradas en la aplicación hasta la próxima sincronización, lo cual constituye una ventaja en la Universidad debido al ritmo dinámico que la caracteriza, además como localmente se tendrá la distribución correspondiente a cada puerta, el proceso de buscar a la persona será más eficiente.

Con el objetivo de lograr el control del acceso, se introduce el número del solapín en el sistema de forma manual o se lee el código de barras impreso en el mismo correspondiente a cada persona. Luego se busca la persona verificando que corresponde a la puerta y se registra el acceso si no ha accedido con anterioridad, en caso contrario, se le informará por cuál puerta accederá.

El **objeto de estudio** de este trabajo es controlar el acceso de personas a puertas de los comedores en la UCI.

El **campo de acción** queda enmarcado específicamente en la parte de control de accesos a comedores de las personas que estudian y trabajan en la Universidad.

El **objetivo general** de este trabajo es desarrollar una aplicación de escritorio que garantice el control de accesos a comedores en la UCI, muestre los datos de cada persona y brinde reportes con la información referente a los comensales por tipo que han accedido en la sesión de alimentación correspondiente.

De aquí, se derivan los siguientes **objetivos específicos**:

Confección de un sistema de control de acceso a comedores que permita:

- Registrar la entrada a los comedores de las personas vinculadas a la UCI
- Actualizar la información de las personas que pasan por cada puerta.
- Mostrar los datos de cada persona

- Brindar reportes sobre la cantidad de personas por tipo que van teniendo acceso.

Para cumplir los objetivos trazados, se desarrollaron las siguientes **tareas**:

- Realizar estudio del entorno de trabajo, identificando las necesidades de la institución.
- Declarar los requisitos que debe cumplir el sistema.
- Describir los procesos que se van a implementar en el sistema.
- Modelar conceptualmente las clases que están implicadas en el sistema.
- Desarrollar los diagramas de actividad.
- Desarrollar los diagramas que describen el diseño del sistema.
- Describir las clases del diseño.
- Diseñar la base de datos local.
- Diseñar la interfaz.
- Implementar la aplicación.

Aportes prácticos:

Con el desarrollo de este sistema se garantiza una herramienta de trabajo capaz de gestionar confiable y eficientemente el acceso de las personas a los comedores, además de mostrar el número de comensales por tipo de personas que van accediendo.

Este trabajo ha sido organizado en cinco capítulos, los cuales se estructuran de la siguiente manera:

Capítulo 1: Incluye un estado del arte del control de acceso a nivel internacional, nacional y de la Universidad, además se exponen las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad; los cuales sirven de base para el desarrollo de la aplicación. Para finalmente darse la propuesta de solución concluyente.

Capítulo 2: Describe cómo se realiza el proceso de control de acceso a comedores en la Universidad actualmente. En él se mencionan los principales problemas que generaron la necesidad del cambio, los procesos que son objetos de automatización, la propuesta del sistema y los requisitos funcionales y no funcionales del mismo.

Describe además el negocio a través de un Modelo de Negocio, las reglas a considerar, los actores y trabajadores que intervienen, los casos de uso del negocio, así como el diagrama de ellos, y el modelo de objetos; también los actores y casos de uso del sistema con su descripción y la representación gráfica del diagrama de casos de uso del sistema.

Capítulo 3: Aborda el análisis y diseño del sistema a desarrollar utilizando herramientas de modelación para mostrar detalladamente los principales procesos que ocurren. Se ilustra la representación gráfica del diagrama de clases de análisis, diagrama de interacción, diagrama de clases y diagrama entidad relación con sus descripciones respectivamente.

Capítulo 4: Trata sobre la implementación, contiene los diagramas de despliegue y de componente, con una breve descripción de sus conceptos.

Capítulo 5: Muestra un estudio de factibilidad para la construcción del sistema, analizando los costos y los beneficios. Se describen los beneficios tangibles e intangibles del desarrollo de la propuesta.

CAPÍTULO 1. Fundamentación Teórica

1.1 Introducción.

En el presente capítulo se ofrece una visión general de los aspectos relacionados con el Sistema de Control de Acceso a Comedores en la UCI. Se describe el desarrollo a nivel internacional y nacional de aplicaciones de control de acceso de personas, además se realiza un estudio de las tendencias, técnicas y tecnologías usadas en la actualidad proponiéndose las utilizadas para el desarrollo de la aplicación.

1.2 Estado del arte nacional e internacional.

1.2.1 Estado del arte nacional.

La Universidad cuenta en la actualidad con un sistema que controla accesos de personas, se trata de:

Sistema de Control de Acceso (UCI)

El Sistema de Control de Acceso tiene entre sus objetivos:

- Controlar el acceso a la universidad, a través de la lectura del código de barras impreso al dorso de los solapines.
- Controlar el acceso diariamente, de las personas que entran y salen por cada una de las puertas de acceso a la Universidad.

Este sistema presta servicios en la actualidad, y es a través de él que se controla el acceso del personal que circula a diario en la Universidad.

1.2.2 Estado del arte internacional.

En el mundo existe gran cantidad de sistemas que de una manera u otra, están relacionados con el control de acceso. Los más conocidos son:

Lantec QuickPass

Es un software diseñado sin complejidades de manejo. Contempla interfaces con periféricos de control como lectores de huellas digitales, tarjetas magnéticas, identificaciones con código de barras, firmas digitales, etc. Emite listados de gestión, históricos por fecha y horario, por personas, grupos o categorías,

etc. Así mismo genera reportes sobre agendas previamente cargadas, y compara lo real con lo planificado, destacando cambios y desvíos.

Sistema Control de Acceso de IBIX

El Sistema de Control de Acceso de IBIX es utilizado primordialmente para restringir el acceso a áreas controladas. Una vez registrado el acceso del personal, sus registros pueden ser utilizados para obtener reportes de asistencia y prenómina.

- **Sistema uControl 6000 Reloj Chequeador Ethernet:**

Este sistema tiene características tales como:

- Capacidad 6,144 trabajadores por reloj
- Memoria cíclica hasta 12,288 registros
- Puerto de red ethernet TCP/IP 10/100BaseT DHCP
- Batería respaldo de datos hasta por 5 años
- Batería respaldo de operación 4 a 8 horas
- Display iluminado 16x2 con mensaje rotativo

Lector a escoger: Código de Barras Láser Lectura Láser, Lector de Banda Magnética (Tarjeta Bancaria Nómina), Lector de Proximidad HID ProxPoint Plus o Lector de Llave Dallas Touch Memory.



Figura 1. Reloj Chequeador uControl 6000 Ethernet.

- **Reloj Chequeador HandPunch**

Es un sistema de Terminal de Reloj Chequeador con Biometría de Mano. El captura una imagen tridimensional de la mano con forma y dimensión para verificar su identidad de una manera práctica y

exacta. Se utiliza en empresas grandes que requieran capturar accesos auténticos de muchos trabajadores.



Figura 2. Reloj Chequeador HandPunch o Terminal Lectora de Mano.

- **Sistema IBIX BioControl 1000/A11Prox:**

Este sistema tiene características tales como:

- Terminal BioControl 1000/A11Prox con Lector de Huella Digital
- Lector Interno de Tarjetas de Proximidad HID
- Capacidad para 1000 huellas digitales
- El control sólo requiere Huella ó Tecleando NIP ó con Tarjeta de Proximidad HID
- Pantalla de cristal líquido retroiluminada
- Teclado Numérico, Reloj de Tiempo Real
- Comunicación RS232, RS485 y Ethernet TCP/IP 10/100BaseT
- Algoritmo de Verificación 1:1 y Autoidentificación 1:N



Figura 3. Reloj Chequeador BioControl 1000/A11.

AMWATCH Access

Este software de control de acceso es una de las principales aplicaciones de los sistemas integrados AMWATCH de seguridad y gestión. La edición estándar ofrece una completa base de datos de usuarios con múltiples campos. Soporta conexiones con sitios a distancia utilizando el protocolo TCP/IP y módems de marcado. Basados en privilegios de usuarios, los operarios pueden definir o modificar parámetros (es decir niveles de acceso), generar reportes, o activar dispositivos controlados.

- Tiene capacidad de manejar hasta 64 paneles, con una capacidad de 128 lectoras.
- La combinación de una arquitectura flexible y abierta usando conectividad TCP/IP o RS-485 y hasta 8 números de operarios.
- Proporciona una base de datos de 20 000 usuarios. Además de 255 zonas de tiempo con 4 períodos.

bioAccess

Es un sistema de control de acceso de personas que utiliza una fuente de identificación confiable basada en la lectura de huellas digitales, diseñado para controlar y administrar las entradas, salidas y movimientos dentro de la institución. Es confiable, rápido y práctico, mejora el control de personas mediante identificación biométrica y está especialmente diseñado para una moderna y segura administración. Posee una efectividad comprobada a 30.000 huellas por segundo, con conexión con cualquier tipo de dispositivo de apertura y es fácilmente integrable a sistemas existentes.

1.3 Tendencias y tecnologías actuales.

1.3.1 Fundamentación de las tecnologías.

Brindar una variada gama de tecnologías de hardware y software es una solución eficaz para las grandes y pequeñas empresas que requieren máxima seguridad, robustez y flexibilidad de programación. Un software de control de acceso, permite llevar el control de horarios y de accesos a lugares específicos. Lo ideal por tanto es hallar esa solución que garantice una correspondencia costo/beneficio y que requiera la menor cantidad de esfuerzo en su puesta en marcha y uso.

1.3.2 Tecnologías de hardware utilizadas para el control del acceso.

Sobre tipos de lectores y tarjetas, ventajas y desventajas de cada uno:

Claves por Teclado

De todas las opciones es la más económica pero menos segura, por lo cual está casi en desuso.

Tarjetas de Banda Magnética

Es de las tecnologías la más conocida, y por consiguiente difundida, utilizada en muchos sistemas de control de acceso. Tienen popularidad y son de muy bajo costo, pero de todos los medios de identificación es el más vulnerable, pues se debe evitar que la banda magnética de la tarjeta se raye o sea expuesta a campos magnéticos que la borren. Son ampliamente usados en sistemas bancarios y de fidelidad.

Touch Memories

Brindan un muy alto nivel de seguridad, pues son altamente resistentes al desgaste, siendo ideales para ambientes industriales. Es una pastilla electrónica, de unos 16 mm. de diámetro y encapsulada en acero inoxidable. No son recomendables para ambientes con alto grado de generación de corriente estática. Son muy confiables, puesto que su tecnología de avanzada evita la posibilidad de duplicarlas.

Tarjetas de Proximidad

Son prácticamente imposibles de duplicar debido a su diseño tecnológico, lo que las convierte en una de las tecnologías más modernas y efectivas. Son prácticas y de bajo costo de mantenimiento, ideales en situaciones de máxima seguridad y alta tecnología, incluso pueden ser leídas dentro de una billetera, una cartera, etc.

Estas tarjetas son llamadas también de Radio Frecuencia (RF) y estos sistemas constan de dos partes; la unidad lectora y la tarjeta. Cuando las tarjetas tienen alimentación interna, se denomina proximidad activa y cuando no tienen esa batería interna se denomina proximidad pasiva, esta última es la más usada. El lector emite una onda electromagnética que genera una corriente interna en la tarjeta, proporcionando su alimentación, dentro de la tarjeta existe un microchip que tiene una memoria que guarda el código y una antena que es la encargada de recibir y transmitir, al inducirse esa corriente en la tarjeta, extrae de la

memoria el dato codificado y lo devuelve, enviándolo al lector, este lo recibe, lo decodifica, lo filtra, lo amplía y lo envía a la unidad de control de accesos a la que esté conectada.

Sistemas Biométricos

Su funcionamiento se basa en la lectura o reconocimiento de alguna parte del cuerpo humano; de la huella dactilar, geometría de la mano, frecuencia de la voz, por la retina o reconocimiento facial; sin necesitar el uso de las tarjetas. De todos, los más conocidos son los lectores de huellas digitales, geometría de la mano e iris del ojo. La principal ventaja de estos sistemas es la seguridad, y su desventaja está relacionada con la velocidad y el precio, además de la poca posibilidad de ser autónomos (generalmente por su complicada lógica trabajan con un software de análisis y una PC conectada directa al lector).

Principales Tecnologías Biométricas	Otras Tecnologías Biométricas
<ul style="list-style-type: none">• Huella Digital (Fingerprint)• Biometría de Mano (Hand Geometry)• Iris	<ul style="list-style-type: none">• Rostro• Voz• Firma• Patrones de Venas

Tabla 1. Tecnologías biométricas.

Características de Tecnologías Biométricas:

- Huella Digital: mide características y patrones de las líneas de las huellas.
- Geometría de Mano: medidas precisas del tamaño y forma de la mano.
- Iris: cámara de video usada para examinar el Iris del ojo humano.
- Rostro: medición de características faciales utilizando una cámara.
- Voz: uso de micrófono para capturar el habla y comparar patrones.
- Firma: se usa una pluma ó stylus para medir la velocidad, presión y dirección, gravando el patrón de una firma hecha a mano.

Tarjetas de Código de Barras

Son aquellas que contienen en sí un código que almacena información y datos que pueden ser reunidos en él de manera rápida y precisa, por lo que representan un método simple y fácil para codificación de información de texto, dicha información puede ser leída por dispositivos ópticos y enviada a una computadora como si hubiese sido tecleada.

El Código de Barras es una disposición en paralelo de barras y espacios que contienen información codificada en las barras y espacios del símbolo. (MARIA, Y, 2005)

Esta tecnología es de fácil implementación y muy económica, puesto que con una impresora pueden ser impresos los carnés de identificación con códigos de barras. Existen varios tipos de codificación, en la actualidad, los más conocidos son:

- Código 39
- Código 39 ASCII Total
- Codabar
- Intercalado 2 de 5
- Código 128
- UPC (Código Universal de Producto)
- EAN (European Article Numbering o Sistema de Numeración Europeo)
- Código 93
- Pharmacode

Beneficios del código de barras

Los beneficios de utilizar códigos de barras como sistema de recolección de datos automática son muy sencillos: la velocidad y exactitud. Una y otra vez se ha probado que capturar datos con códigos de barras es por lo menos 100 veces más rápido y más exacto que la captura normal por teclado, que se traduce en un aumento dramático en la eficiencia y productividad para cualquier operación. (CRUZ, T. L. O. y P. P. ZURITA, 2006)

Tipos de lectores de códigos de barras

Los códigos de barras pueden ser leídos en los sistemas de control de acceso actuales, utilizando diferentes dispositivos. Por lo general consta de dos partes: una interfaz, llamada "decodificador", y lo que se conoce por el término de "dispositivo de entrada".

Tipos de lectores:

- **Sistemas En Línea**

Están conectados de una manera semi-permanente a una computadora o terminal, y su función principal es leer un código de barras, interpretarlo (decodificarlo) y transmitirlo inmediatamente. Por lo general, estos sistemas toman formas de: conexión directa a teclado o "wedge", conexión serial y tarjeta de expansión. La conexión "wedge" se utiliza principalmente en microcomputadoras (IBM PC, Apple y compatibles), así como en terminales de sistemas de cómputo de IBM. El "wedge" provee una conexión directa entre el lector de código de barras y el teclado de la PC o terminal. Los sistemas En Línea son programables, por lo general leyendo comandos codificados en código de barras impresos en el manual que se incluye con estos equipos. Todos los parámetros se almacenan en la memoria no volátil del decodificador, y la única manera de cambiarlos es leyendo nuevos comandos que modifiquen la configuración existente. Con ciertos equipos, actualmente es posible editar, separar o filtrar la información del código de barras antes de enviarlo al sistema de cómputo.

- **Sistemas Fuera de Línea**

Un Lector Portátil de código de barras es una microcomputadora integrada, que incluye un display, teclado para captura de datos a mano, capacidad para descifrar códigos de barras, puerto RS-232 que proporciona acceso para comunicaciones con la PC, baterías y memoria para el almacenamiento de programa y datos. Para recaudo informático de códigos de barras, el sistema puede equiparse con lectores tipo lápiz, CCD o pistolas Láser.

El sistema provee los operadores con indicadores, con las cuales los operadores dan de alta datos por medio de las teclas o lectores ópticos. Los datos se graban, hasta que la terminal portátil los transfiera (por medio de puerto de acceso RS-232) al PC, donde son grabados típicamente en los archivos delimitados de ASCII para la importación fácil de programas como Access, FoxPro, Excel y otras aplicaciones.

Esas terminales en algunos casos contienen programas preinstalados (generalmente para el levantamiento de inventarios) o pueden ser escritos de acuerdo con las necesidades específicas de cada trabajo que desee realizar. Para eso las terminales portátiles cuentan con software generadores de aplicaciones y muchas veces utilizan un lenguaje Basic o C para su programación.

Existen otras terminales que ya poseen el lector integrado, es decir un lector óptico tipo pistola integrado en la parte superior de la terminal. La ventaja de utilizar este tipo de terminal portátil es que el operador puede ejecutar la operación informática de recaudo con una sola mano (vs. dos manos cuando utiliza un terminal portátil sin láser integrado). (MARTÍN, M. A. G, 2005)

Un ejemplo de este lector de código de barras es el Voyager de Metrologic.

1.3.2.1 Tecnología utilizada en la UCI.

En la UCI se decidió utiliza el código de barras Código 3 de 9 o Código 39, que es el más utilizado, pues codifica todos los caracteres ASCII (números, letras, símbolos). Puede ser variable en longitud (permitir cualquier número de dígitos) y ofrece gran seguridad a la lectura. Es de tipo discreto (cada carácter es representado por un grupo específico de barras y espacios) y es usado en aplicaciones de mensajería y paquetería, estados de cuenta, números de servicios, credenciales para control de acceso, etc.



Figura 4. Código 39.

Además el lector de código de barras seleccionado para utilizar fue el Voyager de Metrologic, ya que este es un lector portátil o fuera de línea que puede operar en el modo "manos libres" cuando se sitúa sobre su soporte. Es suficiente presentar el código para que el lector realice automáticamente la lectura. Además es programable para lecturas de corto o largo alcance tanto en el modo automático como manual, con lo que incrementa su eficiencia y productividad.

El Voyager 9520 se ha concebido como un lector muy agresivo con una mayor profundidad de campo y una velocidad de lectura que es casi el doble de la de la serie MS 900 y combina las prestaciones del MS 951 y el MS 961 dentro de un mismo escáner para ofrecer mayor flexibilidad al usuario. Por supuesto este producto mantiene el exclusivo sensor infrarrojo patentado por Metrologic y un sistema de control que permite una activación totalmente automática. (IDENTIFIC-AR, 2006)

Este lector es, hoy en día, uno de los escáneres de códigos de barras de línea única más avanzado.



Figura 5. Lector Portátil de código de barras (Voyager de Metrologic).

1.3.3 Tecnologías de software utilizadas.

Tecnología .Net

El Framework de .NET es un entorno de desarrollo distribuido, en el cual se reúnen en conjunto lenguajes y servicios que simplifican el desarrollo y ejecución de aplicaciones .NET. Además de soporta múltiples lenguajes de programación y, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes: C# (C Sharp), Visual Basic, C++, J# (Java #), Jscript.

Es además, un entorno abierto, adaptable o receptivo a nuevos lenguajes de programación y tecnologías. Se han adaptado un conjunto de lenguajes de desarrollo para la plataforma .NET, tal es el caso de Fortran .NET, Cobol .NET, Python .NET, etc. Otra de las características es que ha sido desarrollado con la pretensión de cumplir con todos los estándares actuales, siguiendo patrones de estandarización ya aprobados.

Obtener un entorno específicamente diseñado para el desarrollo y ejecución del software en forma de Servicios Web es el pilar fundamental de la plataforma .NET. Específicamente cómo desarrollar o instalar estos servicios para que se puedan acceder a ello. La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a la funcionalidad que los Servicios Web proveen.

Es multiplataforma, debido a que posee el CLR (Common Language Runtime, es decir, el Motor Común de Ejecución) que es el centro neurálgico del Framework de .NET encargado de gestionar la ejecución de las aplicaciones, aplicar parámetros de seguridad y ejecutar el denominado recolector de basuras. Un programa .NET podrá ser compilado y ejecutado en cualquier plataforma que incluya un CLR.

Common Language Runtime (CLR)

El Common Language Runtime es el motor en tiempo de ejecución del Framework de .NET, es decir, la parte del "entorno" que se encarga de ejecutar el código de los lenguajes de .NET Framework. La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio (MSIL, Microsoft Intermediate Language). Para generar dicho código el compilador se basa en el Common Language Specification (CLS) que determina las reglas necesarias para crear ese código MSIL compatible con el CLR. Para ejecutarse se necesita de un compilador JIT (Just-In-Time) que es el que genera el código máquina real que se ejecuta en la plataforma del cliente.

La compilación JIT la realiza el CLR a medida que el programa invoca métodos, el código ejecutable obtenido, se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.

Microsoft Intermediate Language (MSIL)

Todos los compiladores que generan código para la plataforma .NET no generan código máquina para CPUs x86 ni para ningún otro tipo de CPU concreta, sino que generan código escrito en el lenguaje intermedio conocido como Microsoft Intermediate Language (MSIL). El CLR da a las aplicaciones la sensación de que se están ejecutando sobre una máquina virtual, y precisamente MSIL es el código máquina de la misma. Es decir, MSIL es el único código que es capaz de interpretar el CLR, y por tanto

cuando se dice que un compilador genera código para la plataforma .NET lo que se está diciendo es que genera MSIL.

MSIL ha sido creado por Microsoft tras consultar a numerosos especialistas en la escritura de compiladores y lenguajes tanto del mundo académico como empresarial. Es un lenguaje de un nivel de abstracción mucho más alto que el de la mayoría de los códigos máquina de las CPUs existentes, e incluye instrucciones que permiten trabajar directamente con objetos (crearlos, destruirlos, inicializarlos, llamar a métodos virtuales, etc.), tablas y excepciones (lanzarlas, capturarlas y tratarlas).

La principal ventaja del MSIL es que facilita la ejecución multiplataforma y la integración entre lenguajes al ser independiente de la CPU y proporcionar un formato común para el código máquina generado por todos los compiladores que generen código para .NET. Sin embargo, dado que las CPUs no pueden ejecutar directamente MSIL, antes de ejecutarlo habrá que convertirlo al código nativo de la CPU sobre la que se vaya a ejecutar. De esto se encarga un componente del CLR conocido como compilador JIT (Just-In-Time) que va convirtiendo dinámicamente el código MSIL a ejecutar, en código nativo según sea necesario.

Lenguaje C Sharp

C# es un lenguaje de programación orientado a objetos, desarrollado como parte de la plataforma .NET. Incluye mejoras derivadas de otros lenguajes como C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones, operadores y fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

C# presenta considerables mejoras e innovaciones en áreas como seguridad de tipos, control de versiones, eventos y recolección de elementos no utilizados (liberación de memoria), seguridad implementada por medio de mecanismos de confianza intrínsecos del código. Incluye un amplio soporte de estructuras, componentes, programación orientada a objetos, manipulación de errores, recolección de basura, etc.

C# contiene las herramientas para definir nuevas clases, sus métodos y propiedades, al igual que la sencilla habilidad para implementar encapsulación, herencia y polimorfismo. Tiene un nuevo estilo de documentación XML que se incorpora a lo largo de la aplicación, lo que simplifica la documentación de clases y métodos.

C# presenta capacidad de control de versiones para facilitar la administración y la implementación de aplicaciones, plena interoperabilidad por medio de los servicios de COM+ 1.0 y .NET Framework con un acceso limitado basado en bibliotecas, y compatibilidad con XML para la interacción con componentes basados en tecnología Web.

El lenguaje proporciona la capacidad de generar componentes de sistema duraderos debido a que tiene total compatibilidad entre el modelo de componentes (COM) y la plataforma para la integración de código existente. Además de que presenta gran robustez y seguridad en el tratamiento de tipos. Posee plena compatibilidad con conceptos de metadatos extensibles y es posible interactuar con otros lenguajes, entre plataformas distintas, y con datos heredados.

Integración en Visual Studio.NET

C# es un lenguaje potente, flexible, moderno y seguro. Sin embargo el mayor sentido práctico de este lenguaje se encuentra en Visual Studio .NET debido a la estrecha integración entre dicho entorno y C#, mucho mayor que la que tiene Visual C++ .NET y equivalente e incluso superior a la de Visual Basic .NET. Gracias a esta integración, los diseñadores de Visual Studio .NET son capaces de generar código C# al crear interfaces de usuario basadas en formularios Windows y formularios Web, producir adaptadores de datos, guiones embebidos en documentos ASP.NET, etc. .

Este lenguaje está pensado con el objetivo de ofrecer capacidades RAD a los desarrolladores de C++ sin sacrificar la potencia y flexibilidad. Esas cualidades RAD son las que encontramos en el entorno de Visual Studio .NET, mientras que el propio lenguaje pone el resto. (MEDINA, D. D. A, 2005)

Visual Studio .Net 2003

Visual Studio .NET es la herramienta que Microsoft ha publicado con el fin de crear aplicaciones para la plataforma .NET desde una interfaz visual basada en ventanas. Este software tiene un rasgo particular al constituir un solo ambiente de desarrollo de aplicaciones para todos los lenguajes.

Visual Studio .NET soporta los nuevos lenguajes .NET: C#, Visual Basic .NET y Managed C++, además de C++. Visual Studio .NET puede utilizarse para construir aplicaciones dirigidas a Windows (utilizando Windows Forms), Web (usando ASP.NET y Servicios Web) y dispositivos portátiles (utilizando .NET Compact Framework).

Visual Studio .NET simplifica el desarrollo de aplicaciones en un ambiente que es una mezcla de lenguajes y lo hace a través de ciertas características como por ejemplo diseñadores visuales para XML, HTML, datos y código del lado del servidor. Visual Studio .NET es capaz de proveer este nivel de integración porque cuenta con las facilidades del *.NET Framework*.

Estas son algunas de las principales características del Visual Studio .NET y sus ventajas:

- Lista de Tareas: organiza las tareas y administra los errores y las advertencias en un solo lugar. Las tareas se leen de los comentarios especializados del código fuente y se presentan en un formato tabular, donde se le hace un *doble-click* para saltar a la sección del código fuente donde está el error o advertencia.
- Explorador de la Solución: brinda una vista jerárquica de una solución dividida en proyectos. Permite la administración de los proyectos involucrados en una solución.
- Explorador de Servidores: administra su computador y otros que están en red, incluyendo recursos como el Servidor de SQL, colas de mensajes, servicios.
- Multi-Monitor: permite un mejor uso del espacio disponible en la pantalla.
- Ayuda Dinámica: hace referencias a la documentación de la ayuda disponible de acuerdo a la materia sobre la cual se esté trabajando. (MEDINA, D. D. A, 2005)

Servicio Web

Un **servicio Web** es un componente de software que se comunica con otras aplicaciones permitiendo el intercambio de datos y codificando los mensaje en XML. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores, accediendo a ellos mediante estándares de Internet muy extendidos, como XML, SOAP y WSDL.

Los servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan en colaboración y en las cuales el software está distribuido en diferentes servidores.

En vez de centrarse en las posibilidades genéricas de portabilidad del código, los servicios Web XML proporcionan una solución viable para habilitar la interoperabilidad de datos y sistemas. Los servicios Web XML utilizan mensajería basada en XML como medio fundamental de comunicación de datos para contribuir a reducir las diferencias existentes entre entornos que utilizan distintos modelos de componentes, sistemas operativos y lenguajes de programación. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

Un servicio Web XML puede ser utilizado internamente por una aplicación o bien ser expuesto de forma externa en Internet por varias aplicaciones. Se puede tener acceso a los servicios Web XML desde una aplicación Web, desde una aplicación de escritorio de Windows u otro servicio Web XML.

En la UCI todas las aplicaciones que se utilizan en la intranet ofrecen o “consumen” servicios Web de otras, es decir, existe una interrelación entre los sistemas de la red para lograr la reutilización y la funcionalidad de estas.

Ventajas de los Servicios Web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.

- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes instituciones ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

Extensible Markup Language (XML)

Los servicios Web XML proporcionan un modo de describir sus interfaces con suficiente detalle para permitir a un usuario construir una aplicación cliente para hablar con ellos. Esta descripción se proporciona generalmente en un documento XML que responde al nombre de documento Web Services Description Language (WSDL).

El lenguaje de marcas extensible es un metalenguaje extensible de etiquetas usado para estructurar información en un documento o en general en cualquier fichero que contenga texto. Es un estándar abierto y libre desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML (Standard Generalized Markup Language) y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML (eXtensible Hyper Text Markup Language, lenguaje extensible de marcado de hipertexto), SVG (Scalable Vector Graphics), MathML (Mathematical Markup Language). Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

XML es un estándar para el intercambio de información estructurada entre diferentes plataformas. Es una tecnología sencilla que tiene a otras tecnologías que la complementan y la hacen mucho más extensible y compatible. Además permite al programador dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de datos o el recorrido de las estructuras corren a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

Web Service Definition Language (**WSDL**)

Es un formato XML que describe el conjunto de mensajes que el servicio Web recibirá o enviará y la forma en que se realiza el intercambio de estos. WSDL nos dice que métodos podemos consumir y nos presenta una descripción de funcionalidad de dichos métodos. WSDL especifica el contenido de un mensaje de petición y el aspecto del mensaje de respuesta en una notación inequívoca.

WSDL describe la interfaz pública a los servicios Web y especifica la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

WSDL se usa a menudo en combinación con SOAP y XML Schema. La notación que utiliza un archivo WSDL para describir formatos de mensajes está basada en el estándar XML Schema lo cual significa que es neutral respecto del lenguaje de programación y que está basado en estándares, lo que lo hace apropiado para describir interfaces de servicios Web XML accesibles desde una amplia variedad de plataformas y lenguajes de programación. Además de describir el contenido de los mensajes, WSDL define dónde está disponible el servicio y qué protocolo de comunicaciones utilizar para hablar con el servicio. Un programa cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL. Esto significa que el archivo WSDL define todo lo necesario para escribir un programa que interactúe con un servicio Web XML. Existen varias herramientas disponibles para leer un archivo WSDL y generar el código necesario para comunicarse con un servicio Web XML.

SQL

Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar entre unos y otras sería realmente complicada a gestionar de no ser por la existencia de estándares que nos permiten el realizar las operaciones básicas de una forma universal.

Nos referimos al SQL (Lenguaje de Consulta Estructurado, o Structured Query Language) que es un lenguaje declarativo y estándar de acceso a bases de datos que explota la flexibilidad y potencia de los

sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos, declarativo de alto nivel por su fuerte base teórica y su orientación al manejo de conjuntos de registros permite una alta productividad en codificación. Hablamos por tanto de un lenguaje normalizado que nos permite trabajar con cualquier tipo de lenguaje en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL, PostgreSQL, SAP Database, Firebird, etc).

El hecho de que sea estándar no quiere decir que sea idéntico para cada base de datos. En efecto, determinadas bases de datos implementan funciones específicas.

Una de sus principales virtudes es su sencillez de uso y versatilidad ya que se puede manejar tanto con unas pocas instrucciones (10 – 12), como con estructuras mucho más complejas de cientos de ellas.

Sistema de gestión de Bases de Datos utilizado

SQLite

SQLite es una librería programada en lenguaje C que implementa un completo motor de base de datos multiplataforma que no precisa configuración. A diferencia de los sistemas de gestión de base de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. La librería SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. Los programas que se enlazan con la librería SQLite tienen acceso a una base de datos SQL, utilizando toda la potencia de este lenguaje.

Su principal característica, es su completo soporte para tablas e índices en un único archivo por base de datos, soporte transaccional, rapidez (unas 2 veces más veloz que MySQL y PostgreSQL), escaso tamaño (unas 25 mil líneas de código C) y su completa portabilidad. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host.

SQLite, no permite múltiples usuarios accediendo en modo escritura a la base de datos, debido a que el mecanismo de bloqueo que utiliza es muy "basto", bloquea toda la base de datos. Así esta librería está

especialmente indicada cuando se requiera de una gran rapidez en las consultas y basta que sólo un único usuario pueda realizar modificaciones.

SQLite, cuenta con una utilidad llamada `sqlite` que nos permitirá ejecutar comandos SQL contra una base de datos SQLite en modo consola. Se pueden crear bases de datos, realizar consultas, insertar datos, etc.

SQLite provee rapidez y simplicidad de la base de datos lo que nos da una gran facilidad de manipulación de los mismos.

1.3.3.1 Metodología a utilizar.

RUP (Rational Unified Process)

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. (BOOCH, G., RUMBAUGH, J., JACOBSON, 2004)

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. (BOOCH, G., RUMBAUGH, J., JACOBSON, 2004)

El Proceso Unificado utiliza el *Lenguaje Unificado de Modelado* (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML, es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos. (BOOCH, G., RUMBAUGH, J., JACOBSON, 2004)

No obstante, los verdaderos aspectos que definen el ciclo de vida del Proceso Unificado de Desarrollo se resumen en tres características fundamentales:

- Dirigido por casos de uso: Los casos de uso reflejan lo que desean y necesitan los usuarios futuros, lo cual se capta cuando se modela el negocio y se representa a través de los

requerimientos. A partir de aquí, los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo, describe los elementos del modelo que son importantes para su construcción y los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- **Iterativo e incremental:** RUP propone que cada fase se desarrolle en iteraciones, estas irán creciendo incrementalmente en cada iteración puesto que se realizarán actividades dentro de los flujos de trabajo que antes no fueron realizadas. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Las iteraciones deben estar controladas; deben seleccionarse y ejecutarse de una forma planificada.

1.3.3.2 Lenguaje de modelado a utilizar.

UML (Unified Modeling Language)

UML (Lenguaje Unificado de Modelado - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el ciclo de desarrollo de un sistema de software. UML ofrece un estándar para describir modelos, incluyendo aspectos conceptuales tales como procesos de negocio y funciones de sistema, además de aspectos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables. UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto. Permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Tiene como objetivo

brindar un material de apoyo que le permita al lector poder definir diagramas propios como también entender diagramas ya existentes. (CRUZ, T. L. O. y P. P. ZURITA, 2006)

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. (CRUZ, T. L. O. y P. P. ZURITA, 2006)

1.4 La propuesta.

La UCI se lleva a cabo una estrategia de migración a software libre de todos los sistemas que se desarrollen, sin embargo el Sistema Control de Acceso a Comedores en la UCI no se desarrollará en esa plataforma por la necesidad urgente de su utilización, ya que es necesario un nuevo sistema que de solución a los problemas que existen actualmente con los accesos de las personas a los comedores.

Tomando como base las herramientas y tecnologías disponibles en la actualidad, para la programación de esta aplicación se utilizó como herramienta el Visual Studio .NET, que permite mediante un único IDE, la elaboración de aplicaciones de escritorio, bases de datos, Servicios Web, hojas de estilo, y otros tipos de componentes; y para modelar dicha aplicación se utilizó el Rational Rose, herramienta líder en el mundo de modelación visual para el proceso de negocio, análisis de requerimientos y diseño de arquitectura de componentes.

Además se utiliza un Servicio Web para el intercambio de información, pues pueden ser utilizados por aplicaciones de software desarrolladas en diferentes lenguajes de programación y ejecutadas sobre cualquier plataforma.

También se usa SQLite para las bases de datos locales que se tendrán en cada puerta de entrada a los comedores porque es muy rápido y su ventaja fundamental es permitir un fácil manejo a los usuarios sin amplio conocimiento del lenguaje, y además no precisa configuración, producto de que el conjunto de la

base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host.

1.5 Conclusiones.

En este capítulo se dio a conocer el estado del arte del tema tratado, haciendo mención de aplicaciones de control de accesos desarrolladas a nivel internacional y nacional. Se mencionaron algunas de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad, quedando expuestas las utilizadas en el diseño e implementación de la aplicación propuesta.

Capítulo 2. Características del sistema

2.1 Introducción.

En el presente capítulo se identifican las necesidades del cliente a través del estudio del negocio, se describen las características del sistema, el flujo actual de los procesos del negocio relacionados con el objeto de estudio que lo soporta a través de un modelo de negocio, los actores y trabajadores que intervienen, los casos de uso del negocio con sus respectivos diagramas y el modelo de objetos. Se especifican además los procesos que serán objeto de automatización y los requisitos principales del software mediante los requerimientos funcionales y no funcionales, definiéndose la propuesta de sistema.

2.2 Objeto de estudio.

El objeto de estudio de este trabajo es controlar el acceso de personas a puertas de los comedores en la UCI.

2.2.1 Problema y situación problemática.

La Universidad de las Ciencias Informáticas (UCI), surgió de la antigua base Lourdes en el año 2002 siendo la primera universidad nacida al calor de la Batalla de Ideas. El principal objetivo de la Universidad es la formación de miles de jóvenes de todo el país como futuros profesionales en la rama de la informática. En ella se garantiza la preparación académica de los estudiantes con la calidad requerida, así como su entrenamiento profesional, a través de la vinculación directa a proyectos productivos; además se les brinda formación y capacitación a aquellos trabajadores que no poseen conocimientos básicos en el manejo de la computadora.

La UCI juega un papel muy importante y decisivo en nuestra sociedad para el desarrollo de la industria nacional del software y del programa de informatización, por lo que se encuentra emergida en el desarrollo de proyectos productivos, específicamente en los sectores de la salud y la educación. Para ello se hace uso de una moderna infraestructura tecnológica y de un método novedoso de organización, con la calidad requerida en las normas internacionales.

Los procesos involucrados al Control de Acceso a Comedores en la UCI tienen lugar del siguiente modo: La persona se presenta en una de las puertas del comedor, muestra su credencial para verificar su acceso (introduciendo manualmente el número de solapín o a través del lector de código de barras) y accede si el proceso es válido, comprobándose que su acceso no se haya registrado con anterioridad. De no ser válido el acceso se mostrará un mensaje para informar el acceso denegado. En caso de que la persona alegue que no tiene su credencial por motivos de pérdida, debe poseer un documento que justifica su entrada firmado y autorizado por el encargado de dicho trámite, sea estudiante, trabajador o visitante. Este autorizo será de uso temporal.

En la actualidad este proceso está muy ligado a las reglas de negocios sobre las que se desarrolló, lo cual implica que al más mínimo cambio haya que hacer arreglos no previstos en el sistema para adecuarlo a las necesidades surgidas, y debido a la urgencia con la que hay que realizarlos, muchas veces los mismos no favorecen su buen funcionamiento. A esto se suma el hecho de que el sistema actual es una aplicación Web, la cual no funciona sin conexión, además de que al no estar conectado directamente a los servidores centrales sino a través de réplicas de los mismos en cada uno de los complejos de comedores, la información que se maneja es redundante y desactualizada. También existen inconvenientes como las demoras de la aplicación en mostrar los datos de las personas que acceden y que las actualizaciones sólo se realizan en determinados momentos.

El proceso de controlar los accesos de personas es imprescindible en la Universidad, y de la forma que está funcionando hoy en día no es capaz de prestar un adecuado servicio dado el incremento del número de comensales y de comedores para dar abasto a ello, siendo este el problema fundamental, pues se requiere de un sistema más dinámico y confiable, apto para las necesidades actuales, y con las mínimas posibilidades de cambios futuros.

2.2.2 Objeto de automatización.

Serán objeto de automatización los procesos que intervienen en el acceso (entrada) de personas que residen (temporales, permanentes) y/o trabajan en la UCI, verificándose su existencia y si tiene autorización para entrar, registrándose cada acceso válido.

En este sistema, los procesos objeto de automatización son: registrar accesos, que significa comprobar que la identificación de dicha persona existe, que puede pasar por esa puerta y que no haya tenido acceso con anterioridad, e informar acceso denegado cuando ha ocurrido algún error; sincronizar accesos, que es actualizar cada cierto tiempo la información que se maneja en las bases de datos locales con las bases de datos centrales a través de un Servidor Web; y brindar un servicio de reportes donde se muestre al usuario la cantidad de personas (por tipo) que han tenido acceso hasta el momento.

En la universidad existen varios sistemas que se relacionan con el campo de acción, como son: el Sistema de Control de Acceso, el cual controla las entradas y salidas a la UCI y el Sistema de Gestión de Credenciales, el cual crea credenciales a personas en la Universidad. Estos sistemas utilizan la tecnología de código de barras para su funcionamiento.

2.2.3 Información que se maneja.

La información que se maneja es la que aparece en la interfaz de la aplicación, en la cual se muestra, en dependencia del tipo de persona, datos tales como nombre, primer y segundo apellidos, número de solapín, foto y tipo de persona; para el estudiante se muestra facultad, grupo y año, y para cualquier otro tipo se muestra el área a la cual pertenece, el código de barras será único para cada persona.

Como documentos específicos que se procesan se tiene, el documento que acredita el acceso si se trata de extravío del solapín, el cual es considerado igual que la credencial, pues incluirá el número del mismo.

2.3 Propuesta de sistema.

La propuesta de solución de este trabajo es desarrollar una aplicación de escritorio que sea capaz de gestionar el control de acceso a comedores en la UCI. Para ello primeramente se introduce el número del solapín en el sistema o se lee el código de barras impreso en el mismo correspondiente a cada persona, se comprobará la existencia de dicha persona y si está autorizada a pasar por la puerta, y luego se verifica si se ha registrado con anterioridad en la misma sesión de alimentación, de existir algún error se devolverá un mensaje que deniegue el acceso, si por el contrario no hay problemas, pues se registrará el acceso y se irá mostrando un reporte de los tipos de comensales que han accedido hasta el momento, esto le servirá de guía a los usuarios de cada una de las puertas.

Otras soluciones que se han encontrado, con gran prestigio a nivel internacional, aunque permiten manejar el proceso de control de acceso en entornos complejos, como es el de la UCI, son muy caras y además no son factibles, sobre todo por la necesidad de integrar el control de acceso a comedores a otros sistemas desarrollados en la Universidad, los cuales le brinden información, por lo que se decidió seguir adelante con la propuesta de solución.

2.4 Modelo de negocio.

Conceptos asociados al modelo del negocio:

Persona: Todo ciudadano perteneciente (vinculado) a la UCI: estudiantes, profesores, trabajador de cualquier área, tercerizados, eventual, familiar, entre otros.

Solapín: Documento identificativo que contiene los datos principales de las personas vinculadas a la UCI.

2.4.1 Reglas del negocio.

Todo proceso de desarrollo tiene ciertas restricciones o reglas que se deben tener en cuenta para modelar o automatizar su comportamiento:

- Ninguna persona que no tenga su solapín, pueda entrar al comedor si no ha sido aprobado por un directivo.
- Cada persona tiene asignado un número de identificación y un código de barras, para identificarlo.
- Sólo el cajero tiene acceso a la información que se genera.
- El cajero tiene que controlar que los accesos no sean reincidentes en la misma sesión de alimentación, e informarlo.

El sistema propuesto permitirá gestionar eficientemente el control de acceso a comedores, pues el mismo será factible y confiable, apto para las necesidades actuales de la Universidad, y con las mínimas posibilidades de cambios futuros. Se garantizará que las distribuciones de personas por puertas se puedan realizar de forma dinámica a través de un Servidor Web, el cual se actualizará con las bases de datos locales cada cierto tiempo, dicha sincronización tendrá lugar de forma manual o automática, siendo posible predeterminar el momento de hacerlo. Además se emitirá un mensaje de error por cada acceso denegado y en caso de que la persona no esté asignada a la puerta por la cual accedió, se le mostrará por dónde deberá acceder.

Para lograr una aplicación que responda a los requerimientos del cliente es importante comprender el negocio actual en el entorno de la Universidad. De este modo se lleva a cabo el **modelo de negocio**, pues en el se comprende cómo funcionan los procesos hasta el momento.

2.4.2 Descripción de actores y trabajadores.

Actores del negocio	Justificación
Persona	Persona que trabaja o estudia en la UCI, tiene un solapín con sus datos personales, puede vivir dentro o fuera de la UCI y necesita autenticarse para entrar al comedor.

Tabla 2. Actores del negocio.

Trabajadores del negocio	Justificación
Cajero	Cajero que radica en la puerta de entrada de cada comedor de la UCI. Es el encargado de pedirles a las personas que entran al comedor de la UCI sus solapines, garantizando su correcto acceso y prohibiendo las reiteraciones en la misma sesión de alimentación. Se encarga de velar por la validez de todo el proceso.
Directivo	Persona trabajadora de la UCI con cargos en la misma ya sea en la dirección de las Facultades de la Universidad o como directivo del comedor; puede autorizar la entrada de las personas al comedor mediante un papel de autorización.

Tabla 3. Trabajadores del negocio.

2.4.3 Diagrama de Casos de Uso del Negocio.

El diagrama de casos de uso del negocio tiene una gran importancia porque representa como interactúan los actores y casos de uso del negocio. Sirve de ayuda al equipo de desarrollo para comprender como funciona el proceso de negocio para posteriormente obtener los casos de uso del sistema.



Figura 6. Diagrama de Casos de Uso del Negocio.

2.4.4 Descripción de casos de uso del negocio. Anexo 1

2.4.5 Diagramas de Actividad. Anexo 1

2.4.6 Modelo de Objetos. Anexo 1

2.4.7 Listado de los requerimientos funcionales.

De acuerdo con los objetivos planteados el sistema debe ser capaz de:

R1- Autenticar usuario: Permitir al usuario acceder a la información que le corresponde.

R1.1- Comparar Usuario y contraseña con los usuarios del sistema.

R2- Registrar las entradas (accesos) de las personas al comedor.

R2.1- Verificar si el acceso es válido por la puerta correspondiente.

R2.2- Verificar si el acceso no se ha registrado con anterioridad.

R3- Configurar el sistema.

R3.1- Configurar tiempos de sincronización de la base de datos.

R3.2- Sincronizar las personas y los accesos de las personas al comedor, o sea, que se actualice el Servidor Web con las bases de datos locales.

R3.3- Mostrar datos de configuración del sistema.

R4- Mostar datos.

R4.1- Mostrar Foto, Nombre, primer y segundo Apellidos, Número de Solapín y Tipo de Persona; si es estudiante, mostrar Facultad, Grupo y Año; para cualquier otro caso mostrar el Área a la cual corresponde.

R4.2- Mostrar mensaje (si existe conexión) para los comensales no incluidos en la distribución pero sí autorizados.

R5- Mostrar reporte del número de comensales por tipo accedidos en la sesión de alimento.

2.4.7.2 Definición de los requerimientos no funcionales.

- Apariencia o interfaz externa

La interfaz externa de la aplicación presentará un diseño entendible, sencillo y fácil de usar, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

La aplicación usará el protocolo HTTP para intercambiar información con el Servidor Web, mientras que para acceder al repositorio de fotos usará el protocolo TCP/IP.

- Usabilidad

El sistema podrá ser usado por cualquier persona (usuarios finales) que posea conocimientos básicos en el manejo de la computadora y esté vinculado al campo de trabajo. La interfaz mostrará únicamente funcionalidades referentes a las tareas para las que está destinado el sistema y se garantiza que los usuarios sólo tengan acceso a la información que les corresponda.

- Rendimiento

El sistema operará con grandes volúmenes de información, por tanto, se hace necesario tiempos de respuestas menores a 5 segundos, al igual que la velocidad de procesamiento de la información, para evitar demoras y grandes colas en los puntos de acceso al comedor (las puertas), garantizando rapidez en los horarios de alimentación. Se debe garantizar la consistencia y disponibilidad de la información en todo momento, por lo que se requiere además un tiempo de recuperación mínimo. Requiere como recurso tecnológico un lector de código de barras por cada PC cliente.

- Soporte

El sistema debe ser de fácil instalación y mantenimiento.

Se requiere de:

Plataforma .NET versiones 1.0 / 1.1 / 2.0.

Un Servicio Web con las siguientes características:

- Soporte para velocidad de procesamiento de los datos.
- Tiempo de respuesta rápido.

- Portabilidad

El sistema no es portable, pues sólo corre sobre Windows.

- Seguridad

El sistema tiene que regirse por los siguientes aspectos:

- Evitar que el usuario pueda realizar cualquier acción sobre la configuración del sistema para evitar accesos no autorizados y divulgación.
- Garantizar que la información sea vista únicamente por quien tiene derecho a verla.
- Protección contra acciones no autorizadas y estados inconsistentes que puedan afectar la integridad de los datos.
- Garantizar que el sistema funcione correctamente aún cuando no haya conectividad.
- Garantizar el acceso a la información por los usuarios autorizados y que los mecanismos de seguridad no oculten, ni retrasen los datos deseados.
- Proporcionar un fichero para registrar los errores del sistema.
- En los clientes sólo se accederá a la base de datos local a través del sistema, nunca se administrará directamente.

- Legales

El sistema debe cumplir con los reglamentos y estándares de software del centro.

- Confiabilidad

- El sistema de gestión de bases de datos tiene soporte para recuperación ante fallos y errores.
- Todas las salidas del sistema tienen que tener el 100% de veracidad y precisión.
- El sistema deberá ser capaz de recuperarse rápidamente de las fallas, en un tiempo promedio de 2 minutos aproximadamente.
- El sistema deberá tener un 100% de disponibilidad para usarlo efectivamente.

- Interfaz interna

Diseño sencillo y fácil de usar, amigable, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

- Software

- Plataforma .NET 1.0 ó posterior instalada en las puertas para la aplicación de escritorio y en el servidor, para el sitio Web.
- SQLite para las puertas.

- Hardware

Para la plataforma .NET

- Procesador Intel Pentium a 133 MHz o superior.
- Sistema operativo Microsoft Windows® 2000.
- Requisitos mínimos de RAM 256 MB (se recomiendan 512 MB o más).
- Espacio en disco duro necesario para la instalación: 600 MB.
- Espacio necesario en el disco duro: 370 MB.

- Restricciones del diseño

Para el análisis y el diseño del sistema se debe utilizar la metodología RUP, usando el lenguaje de modelación UML y como herramienta para llevarlo a cabo el Rational Rose.

2.5 Modelo de Casos de Uso del Sistema.

Los casos de uso del sistema son los encargados de guiar todo el proceso de desarrollo del sistema a implementar. Es importante determinar cuáles serán los actores del sistema, así como los casos de uso del mismo.

2.5.1 Definición de los actores del sistema a automatizar.

Actores	Justificación
Cajero	Cajero que radica en la puerta de entrada de cada comedor de la UCI. Se encarga de garantizar que sea correcto el acceso y que no ocurra repetidas veces en la misma sesión de alimentación. Vela por la validez de todo el proceso.
Administrador del sistema	Es el encargado de controlar la configuración del sistema, ya sea para cambiar los tiempos de sincronización de las personas y de los accesos de dichas personas, o ejecutar la acción de sincronizar por la puerta correspondiente.
Reloj	Actor genérico para indicar un temporizador del cliente, el cual ejecuta ciertas tareas programadas del sistema.

Tabla 4. Actores del Sistema.

2.5.2 Listado de casos de uso del sistema.

Se presenta un listado de los casos de uso del sistema con una breve información de quien es el actor que lo inicializa, el nombre del caso de uso y una breve descripción del mismo.

CU-1	Registrar_Acceso
Actor	Cajero
Descripción	El sistema registra aquellos accesos que son válidos, o sea, los correspondientes a las personas autorizadas por esa puerta y que no hayan accedido con anterioridad; el cajero permite el acceso según la información mostrada por el sistema.
Referencia	RF2, RF2.1, RF2.2

Tabla 5. Descripción del Caso de Uso del Sistema “Registrar_Acceso”.

CU-2	Sincronizar_Persona
Actor	Sincronizador
Descripción	Después de activada la aplicación, tiene lugar la sincronización de forma manual o automática, siendo posible predeterminar el momento de hacerlo, entonces se actualiza el listado de las personas de las bases de datos locales con el Servidor Web.
Referencia	RF3.2

Tabla 6. Descripción del Caso de Uso del Sistema “Sincronizar_Persona”.

CU-3	Sincronizar_Acceso
Actor	Sincronizador
Descripción	Después de activada la aplicación, tiene lugar la sincronización de forma manual o automática, siendo posible predeterminar el momento de hacerlo, entonces se actualizan remotamente en el Servidor Web los accesos que fueron registrados en las bases de datos locales.
Referencia	RF3.2

Tabla 7. Descripción del Caso de Uso del Sistema “Sincronizar_Acceso”.

CU-4	Mostrar_Datos
Actor	Cajero
Descripción	El sistema muestra de cada persona existente, datos como nombre, primer y segundo apellidos, número de solapín, foto y tipo de persona; para el estudiante se muestra facultad, grupo y año, y para cualquier otro tipo se muestra el área a la cual corresponde.
Referencia	RF4, RF4.1, RF4.2

Tabla 8. Descripción del Caso de Uso del Sistema “Mostrar_Datos”.

CU-5	Mostrar_Reportes
Actor	Cajero
Descripción	El sistema muestra un reporte temporal del número de comensales por tipo accedidos en la sesión de alimentación, incrementándose en la medida en que se registran.
Referencia	RF5

Tabla 9. Descripción del Caso de Uso del Sistema “Mostrar_Reportes”.

CU-6	Autenticar
Actor	Usuario
Descripción	El usuario se autentica para poder acceder a la configuración del sistema con el objetivo de realizar alguna acción sobre el mismo.
Referencia	RF1, RF1.1

Tabla 10. Descripción del Caso de Uso del Sistema “Autenticar”.

CU-7	Mostrar_Datos_Configuración
Actor	Administrador del sistema
Descripción	El administrador del sistema puede ver los datos de la configuración del sistema tales como: el último apagado de la aplicación, la última fecha de sincronización de las personas o del acceso de las personas, próxima fecha/hora de sincronización y el tipo de apagado de la aplicación.
Referencia	RF3, RF3.3

Tabla 11. Descripción del Caso de Uso del Sistema “Mostrar_Datos_Configuración”.

CU-8	Actualizar_Datos_Configuración
Actor	Administrador del sistema
Descripción	El administrador del sistema puede actualizar o cambiar parámetros de configuración tales como tiempo de sincronización para las personas o para los accesos de dichas personas, además puede actualizar la puerta correspondiente.
Referencia	RF3.1

Tabla 12. Descripción del Caso de Uso del Sistema “Actualizar_Datos_Configuración”.

2.5.3 Diagrama de casos de uso del sistema a automatizar.

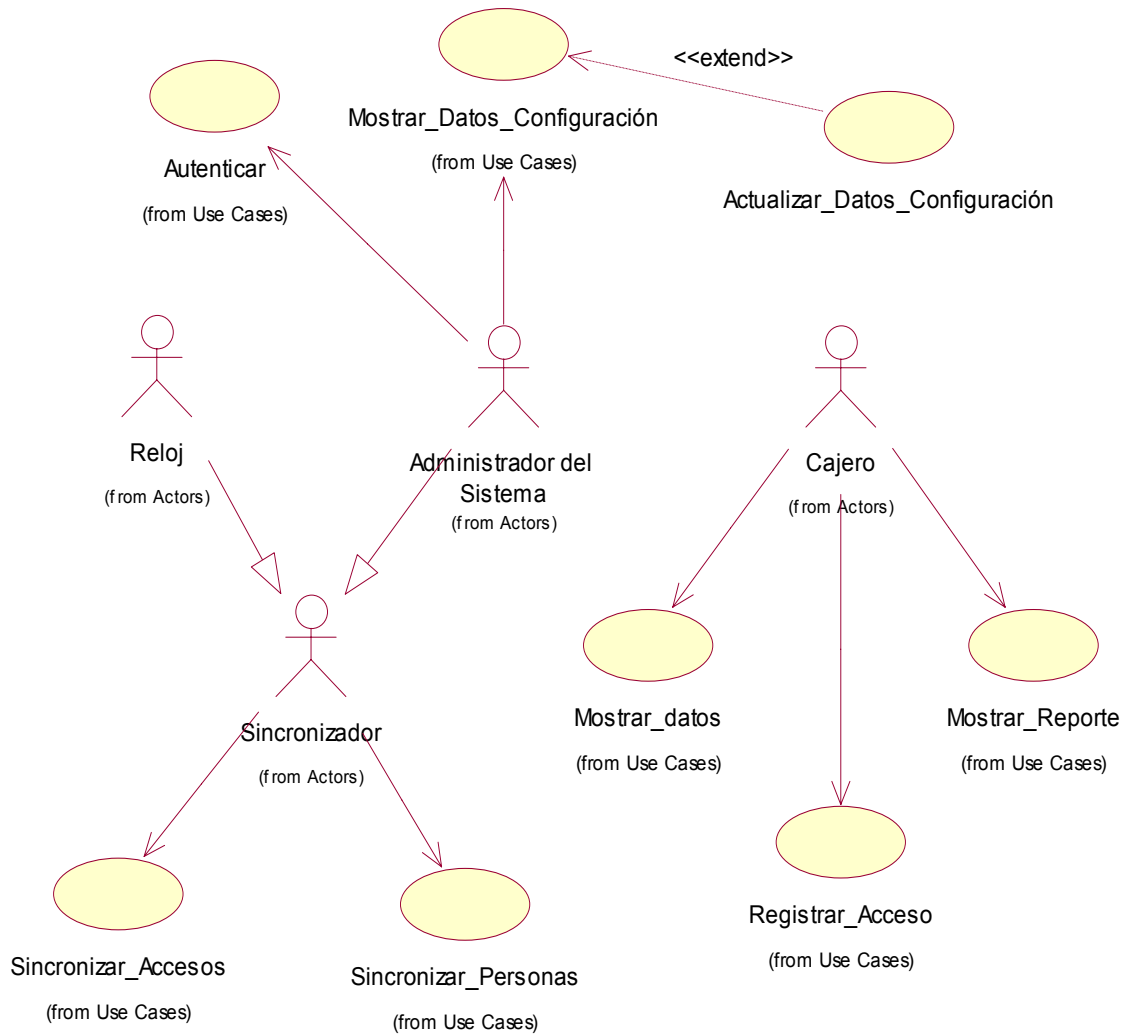


Figura 7. Diagrama de Casos de Uso del Sistema.

2.5.4 Descripción de los casos de uso del sistema. Anexo2

2.6 Conclusiones.

En este capítulo se comenzó el desarrollo de la propuesta de solución que inicialmente se describió, analizándose los procesos de negocio, los requisitos funcionales y no funcionales que tiene el sistema, llegando a los casos de uso necesarios para satisfacer estos requisitos. Con el objetivo de comenzar el análisis y diseño del sistema, guiado por los casos de uso.

Capítulo 3. Análisis y diseño del sistema

3.1 Introducción.

En este capítulo se modelan los artefactos necesarios para el análisis y diseño del sistema. Esto constituye el pilar fundamental para el desarrollo de la aplicación. Se muestran además el diagrama de clases de análisis, el de clases del diseño, y los diagramas de secuencia, también se describen las clases del diseño y se representan el diagrama de clases persistentes y el modelo de datos.

Se describen y muestran las principales interfaces de la aplicación, así como se menciona el tratamiento de errores.

3.2 Análisis.

El análisis brinda una visión muy general de lo que debe hacer el sistema. El modelo del análisis proporciona una vista interna del sistema que está descrita en el lenguaje del desarrollador. A continuación se expone el diagrama de clases del análisis del sistema propuesto.

3.2.1 Modelo de clases de análisis.

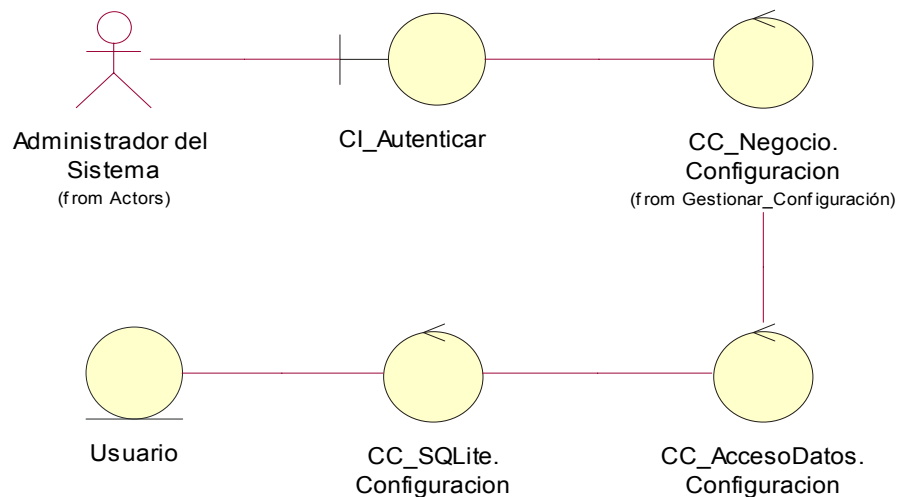


Figura 8. Diagrama de Clases de Análisis del Caso de Uso "Autenticar".

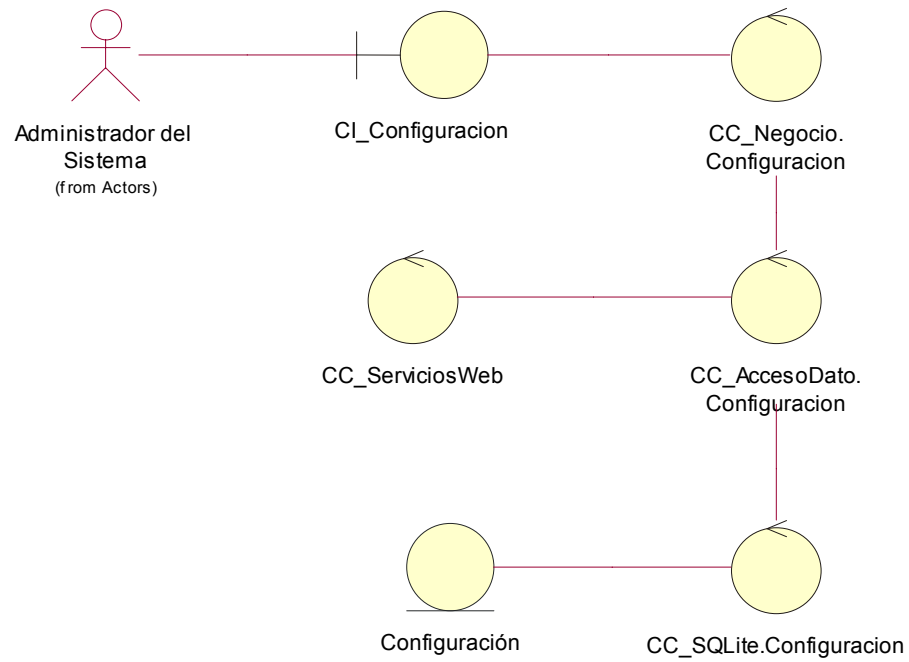


Figura 9. Diagrama de Clases de Análisis del Caso de Uso “Mostrar_Datos_Configuración”.

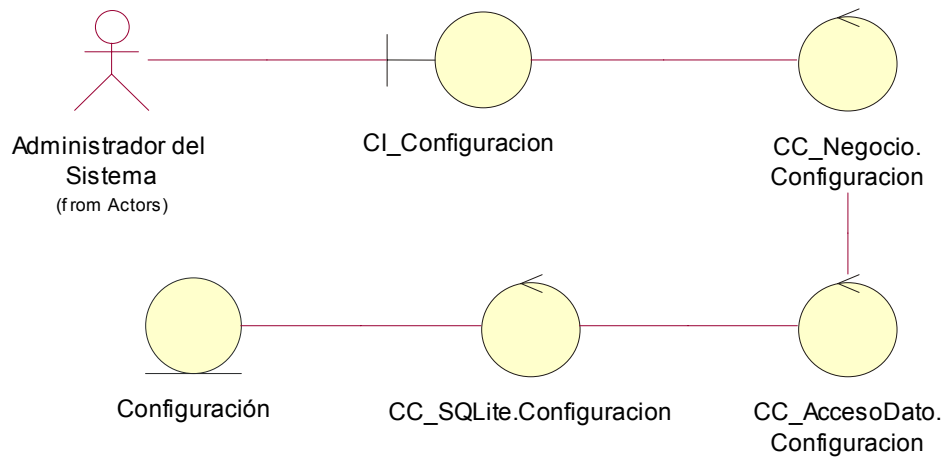


Figura 10. Diagrama de Clases de Análisis del Caso de Uso “Actualizar_Datos_Configuración”.

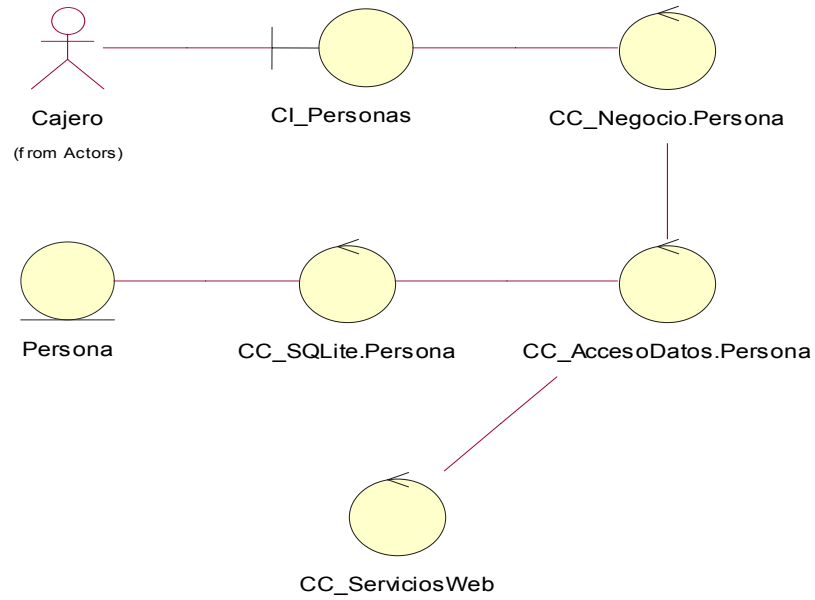


Figura 11. Diagrama de Clases de Análisis del Caso de Uso “Mostrar_Datos”.

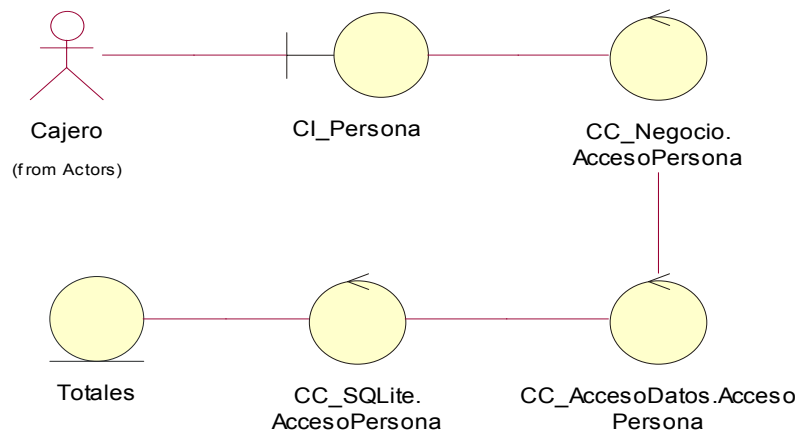


Figura 12. Diagrama de Clases de Análisis del Caso de Uso “Mostrar_Reporte”.

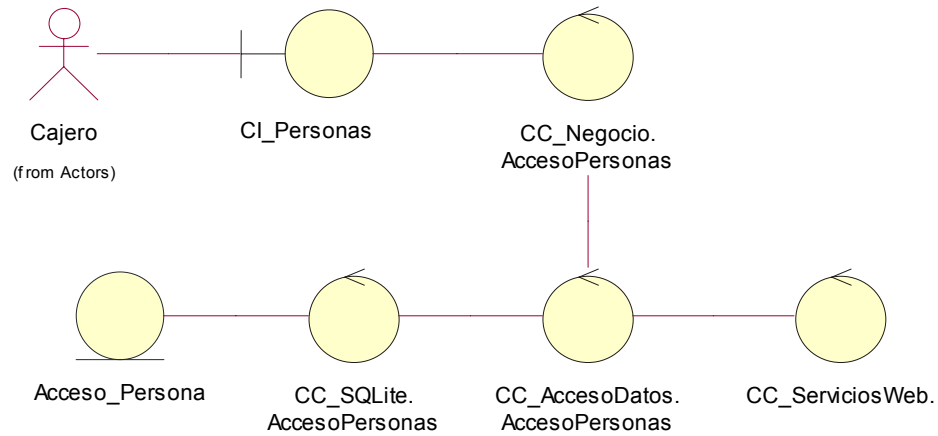


Figura 13. Diagrama de Clases de Análisis del Caso de Uso “Registrar_Acceso”.

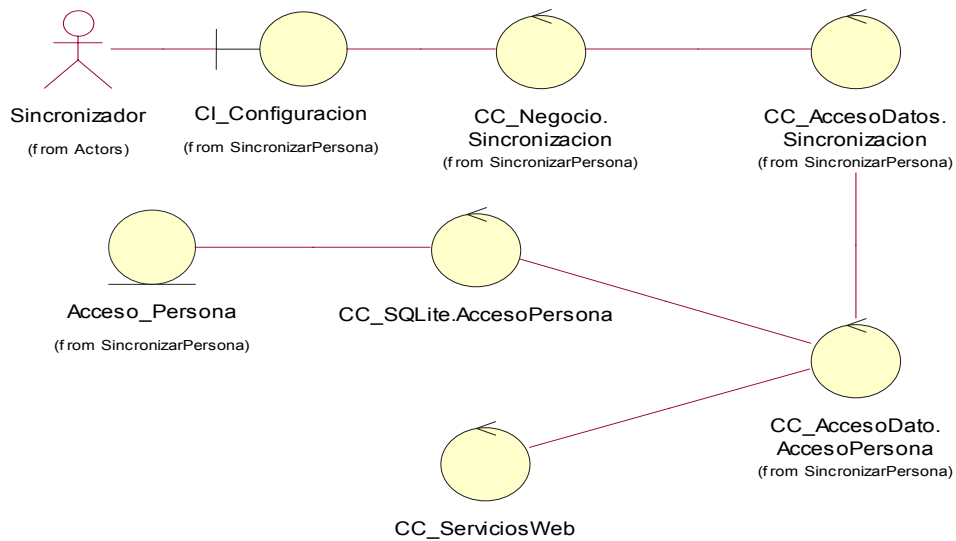


Figura 14. Diagrama de Clases de Análisis del Caso de Uso “Sincronizar_Acceso_Persona”.

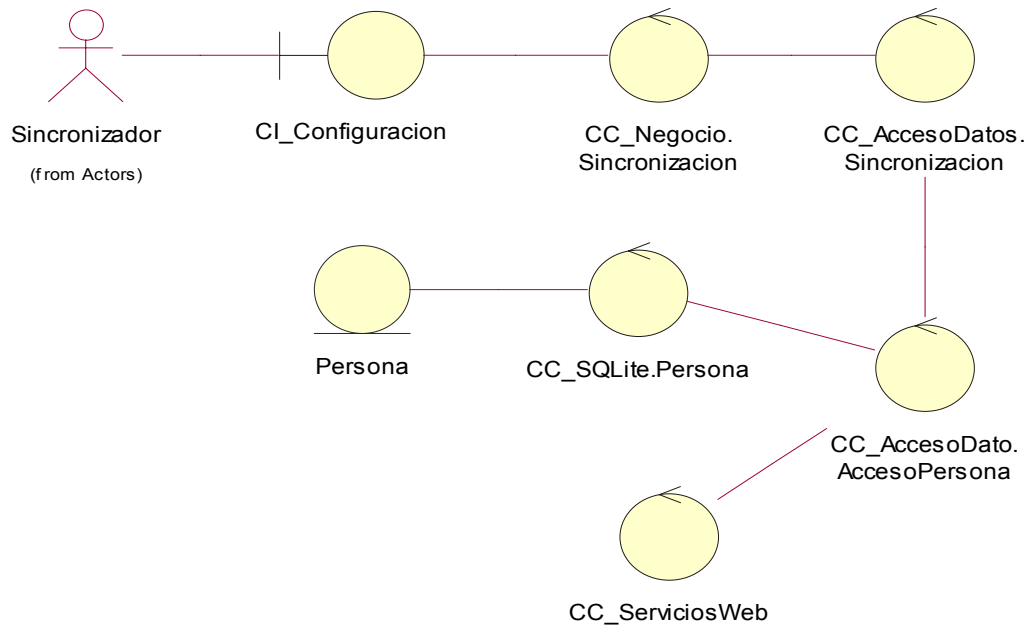


Figura 15. Diagrama de Clases de Análisis del Caso de Uso “Sincronizar_Persona”.

3.3 Diseño.

Los diagramas de clase del diseño serán representados para cada caso de uso. Además serán descritos de forma textual sus métodos y atributos, y para representar el aspecto dinámico de cada uno de ellos, serán utilizados los diagramas de secuencia, los cuales representan la interacción entre clases.

3.3.1 Diagramas de interacción. Anexo 3

3.3.2 Diagrama de clases del diseño.

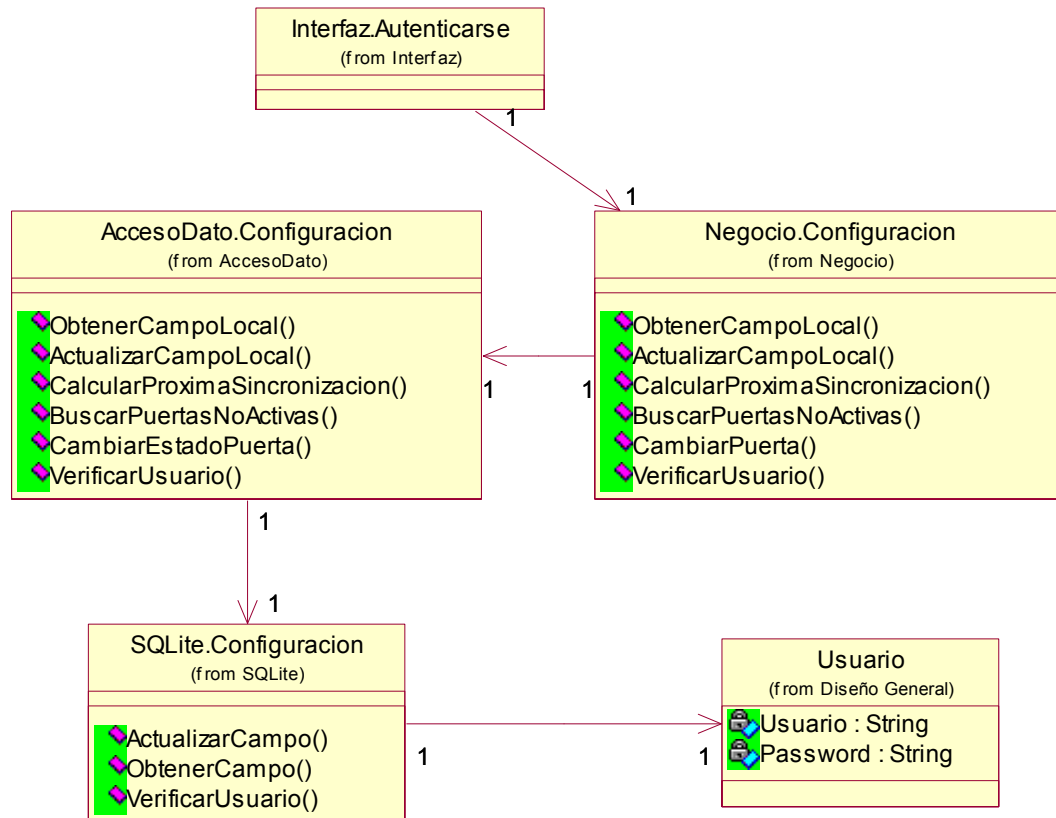


Figura 16. Diagrama de Clases de Diseño del Caso de Uso “Autenticar”.

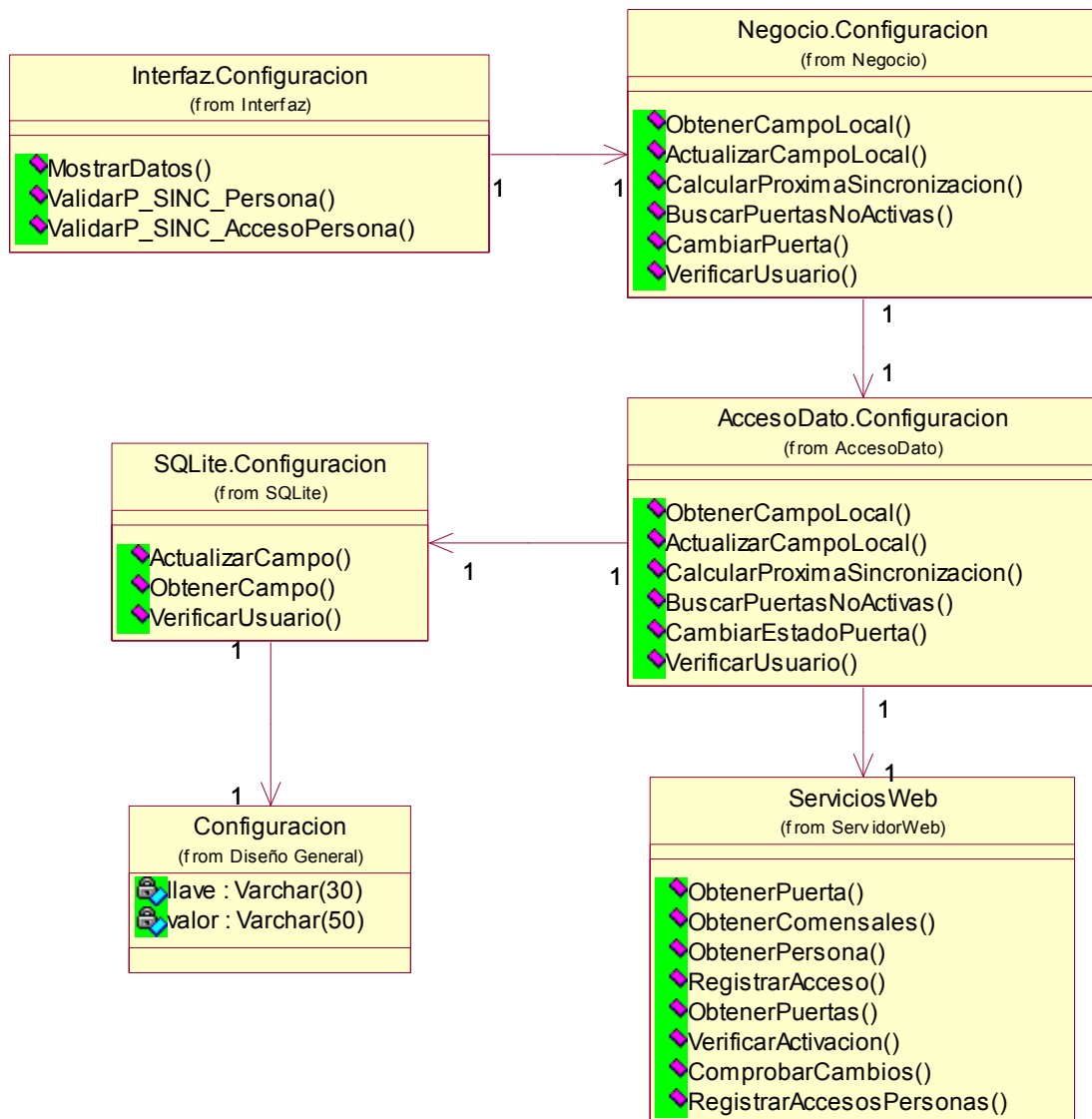


Figura 17. Diagrama de Clases de Diseño del Caso de Uso “Mostrar_Datos_Configuración”.

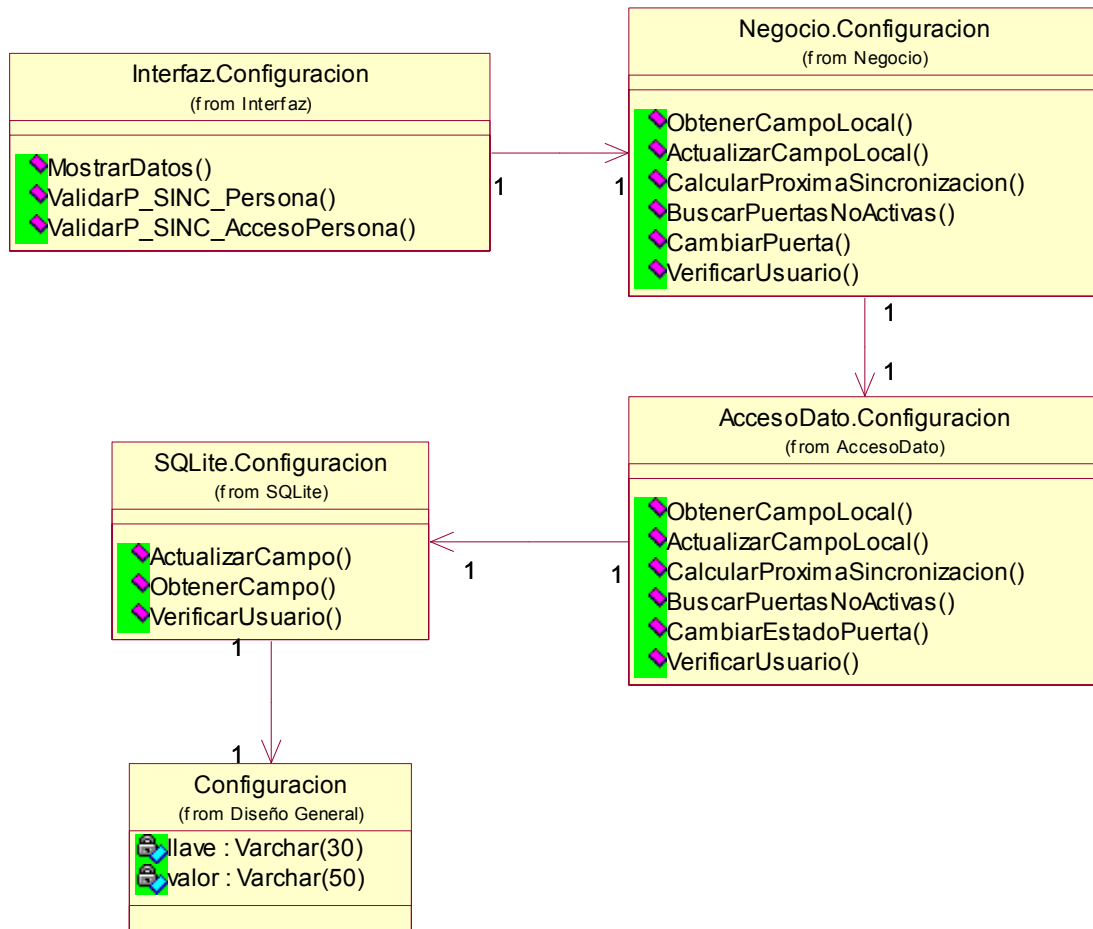


Figura 18. Diagrama de Clases de Diseño del Caso de Uso “Actualizar_Datos_Configuración”.

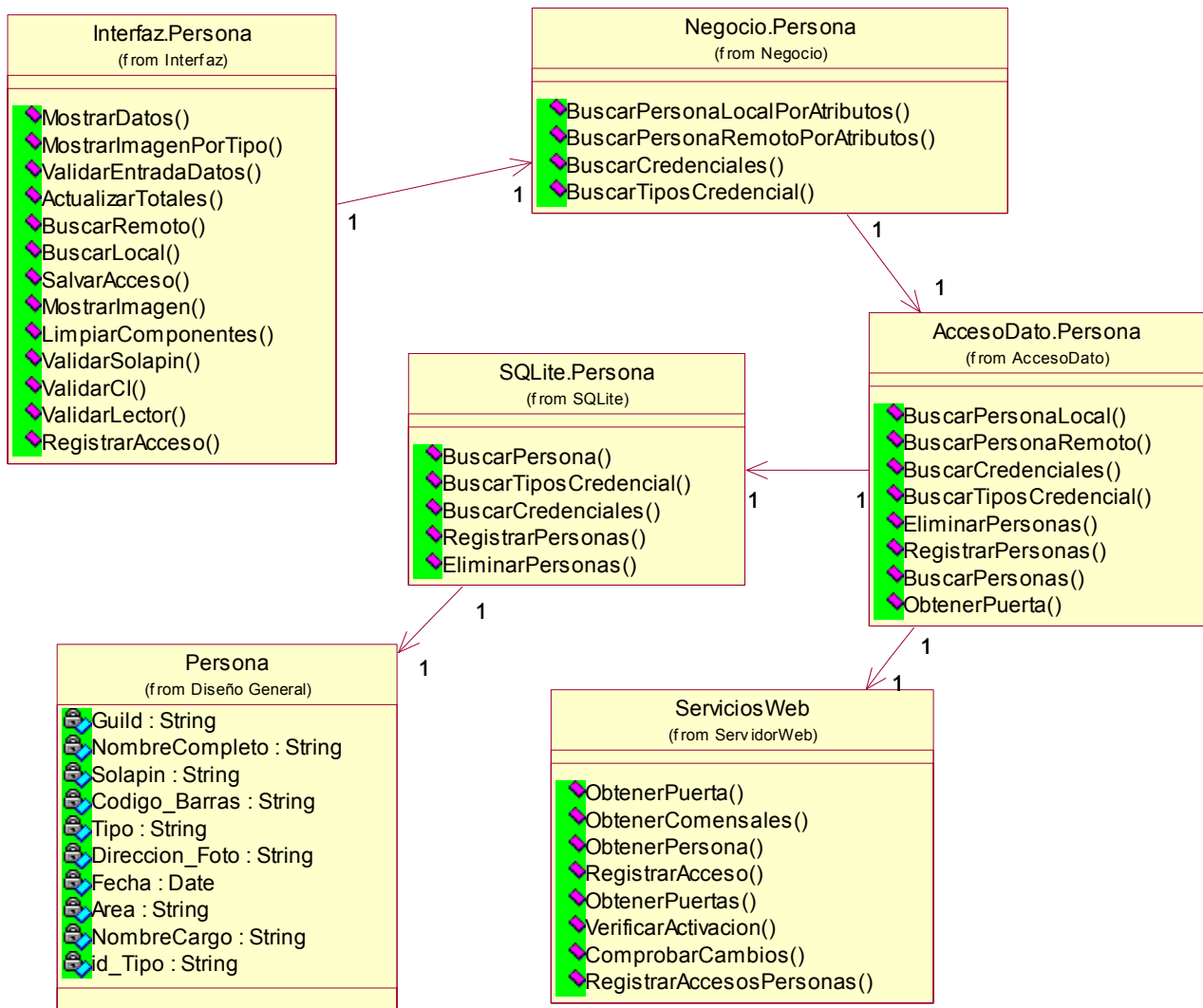


Figura 19. Diagrama de Clases de Diseño del Caso de Uso “Mostrar_Datos”.

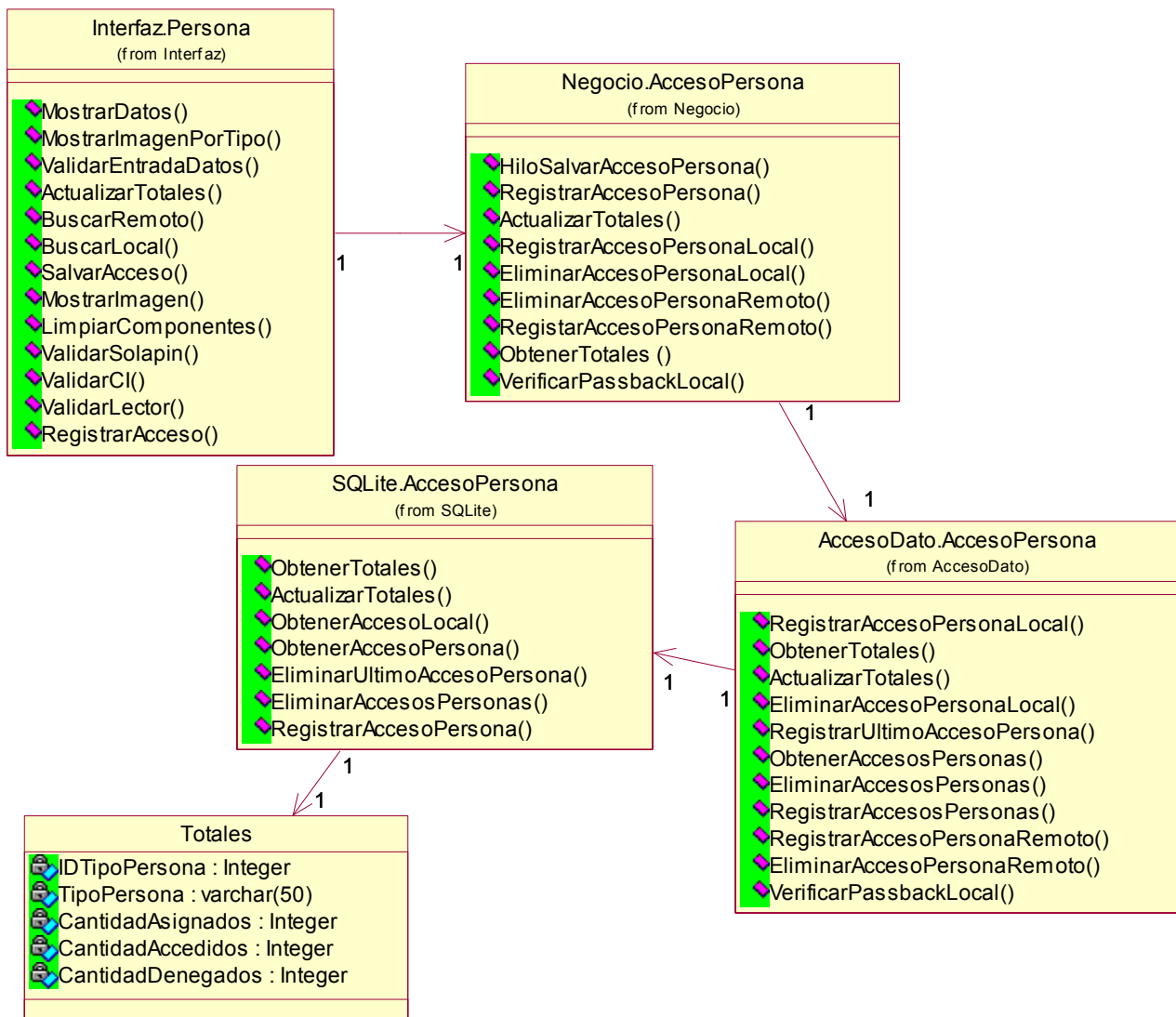


Figura 20. Diagrama de Clases de Diseño del Caso de Uso “Mostrar_Reporte”.



Figura 21. Diagrama de Clases de Diseño del Caso de Uso “Registrar_Acceso”.



Figura 22. Diagrama de Clases de Diseño del Caso de Uso “Sincronizar_Acceso_Persona”.

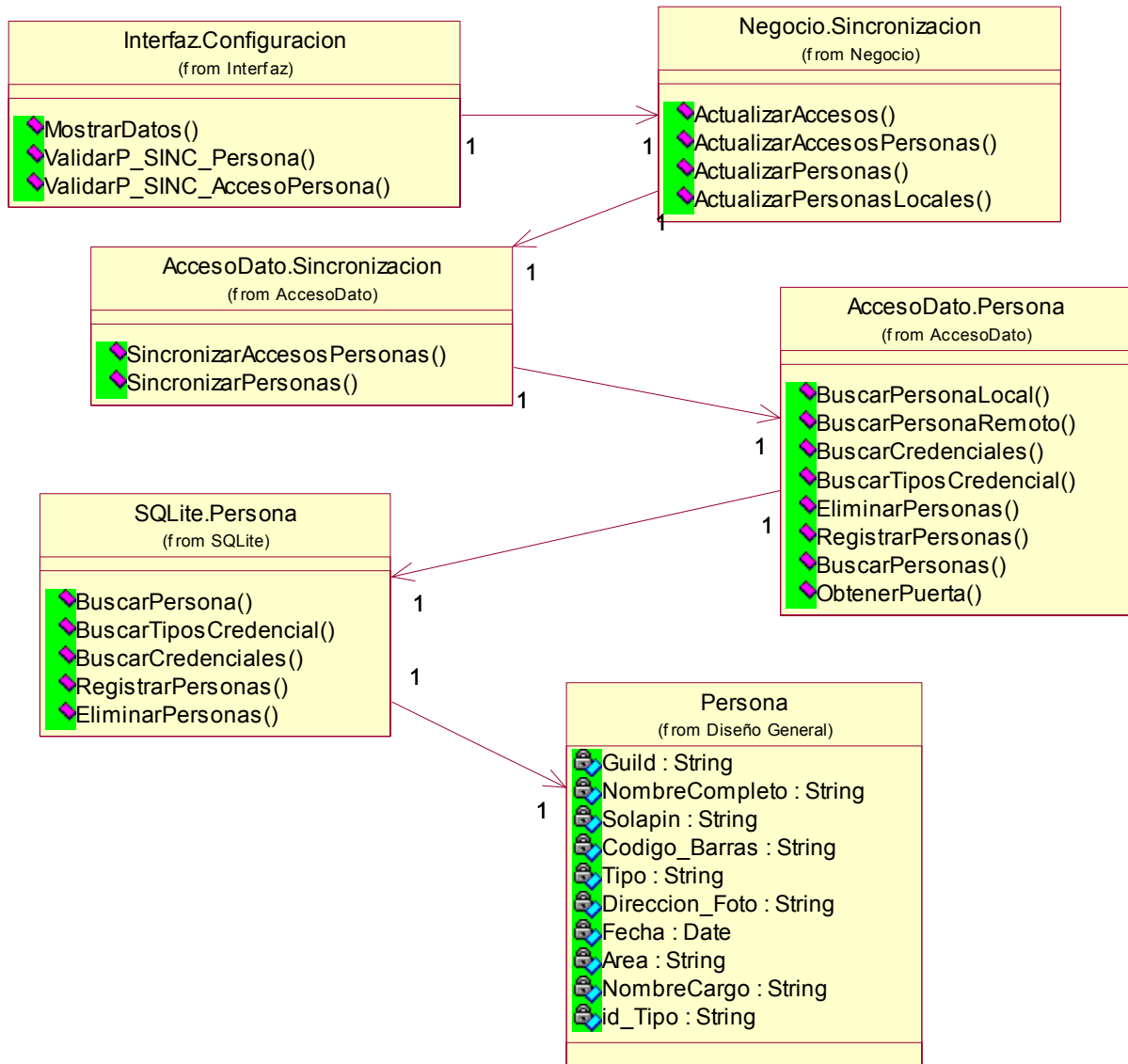


Figura 23. Diagrama de Clases de Diseño del Caso de Uso “Sincronizar_Persona”.

3.4 Diseño de la base de datos.

El almacenamiento de la información es uno de los aspectos más importantes para cualquier aplicación, a continuación se modela la base de datos que será utilizada en el sistema propuesto.

3.4.1 Modelo lógico de datos (diagrama de clases persistentes).

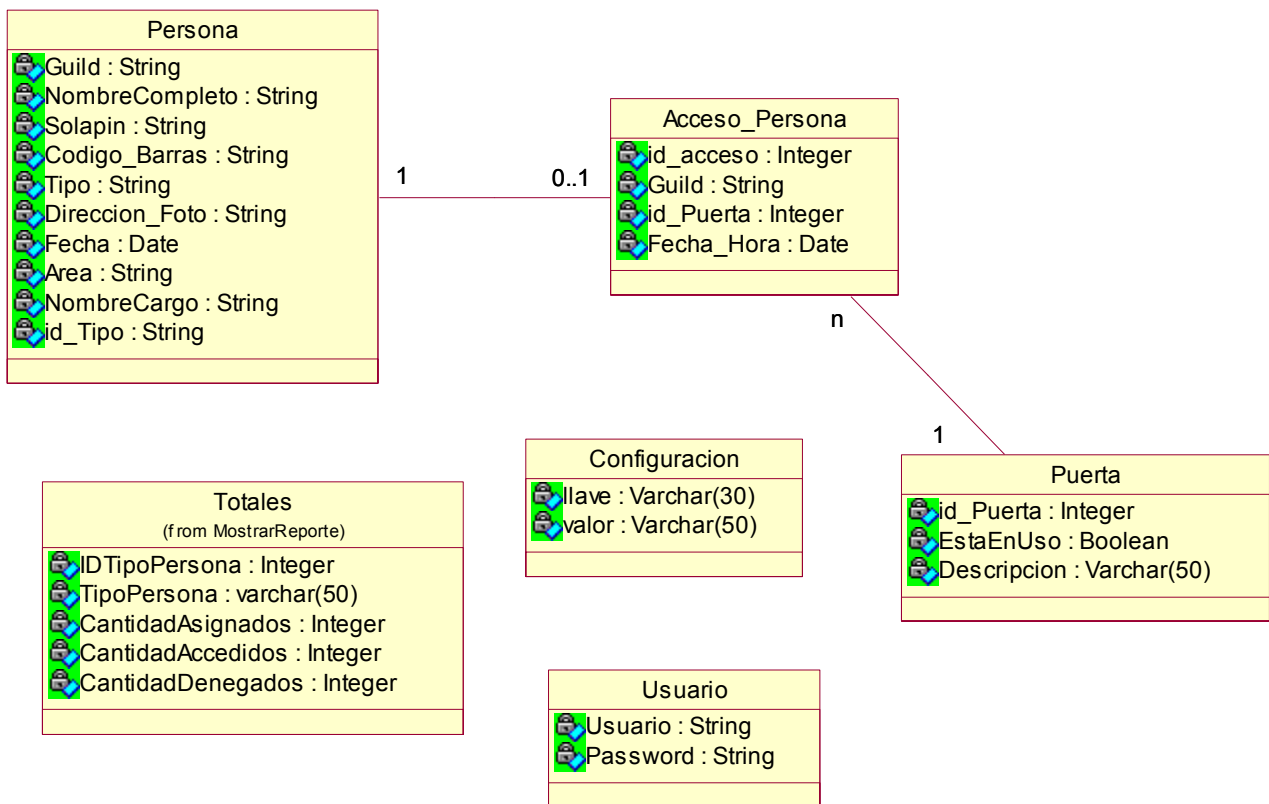


Figura 24. Diagrama de Clases Persistentes.

3.4.2 Modelo físico de datos (modelo de datos).

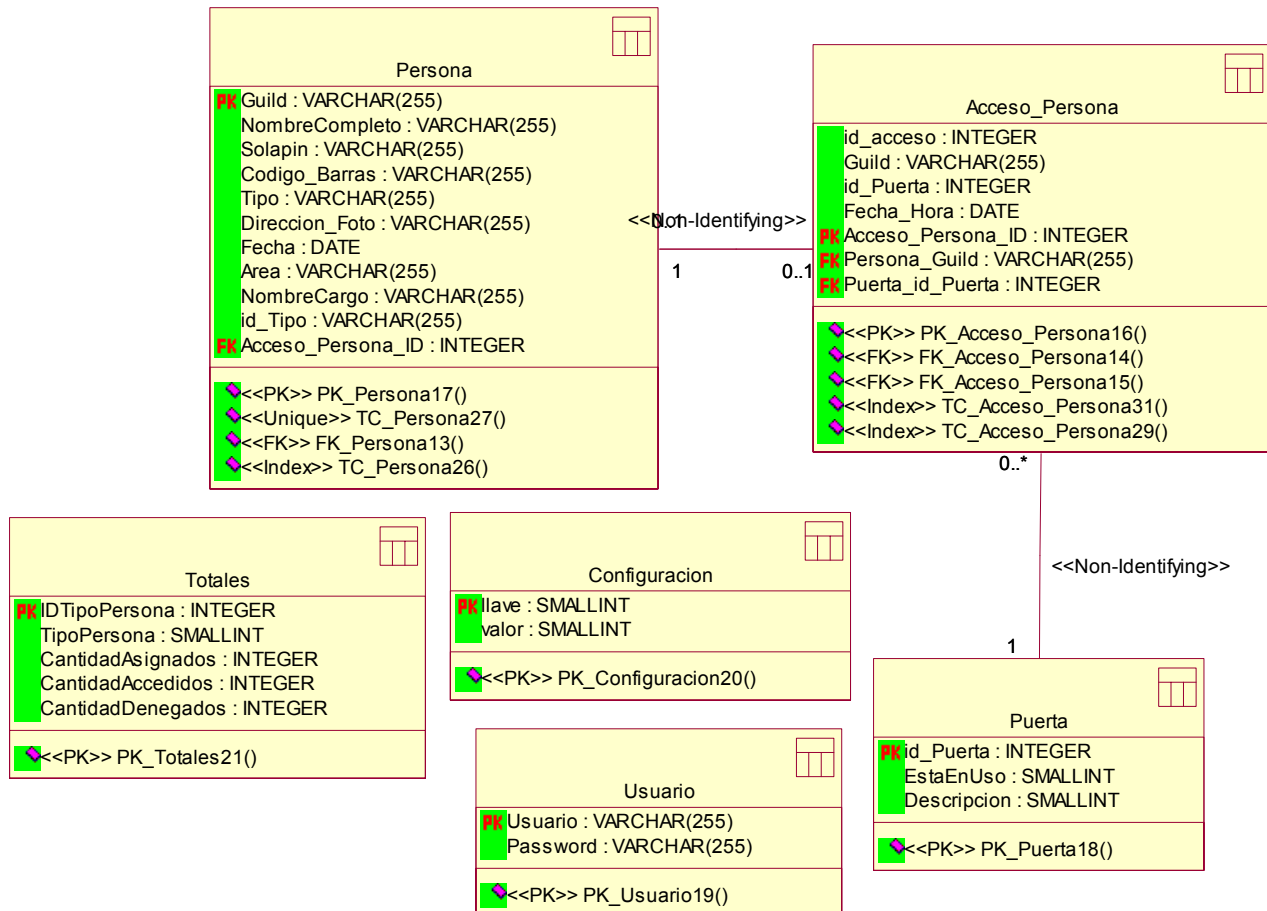


Figura 25. Modelo de Datos.

3.4.3 Descripción de las tablas. Anexo 4

3.5 Arquitectura de Capas.

La arquitectura de una aplicación es la vista conceptual de la estructura de la misma. La de esta aplicación se corresponde con la arquitectura o patrón de 3 capas. La aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos, y dichas capas están asociadas a estos códigos:

Capa de Presentación: La capa de presentación contiene los Componentes de Interfaz de Usuario que son las clases de presentación del sistema, en este caso, se refiere a una aplicación de escritorio.

Capa de Negocio: En esta capa se encuentran ubicados los componentes que se encargan de gestionar toda la lógica del negocio de la aplicación. Está integrada por las clases controladoras que manipulan toda la configuración del sistema, controlan la actualización de las personas y los accesos de personas y registran los accesos locales y remotos de dichas personas.

Capa de Acceso a Datos: A esta capa pertenecen los componentes lógicos de acceso a datos que acceden a los almacenes de datos. Ellos se encargan de realizar todas las operaciones sobre la información propia del sistema. Esta capa contiene las clases que hacen posible la persistencia y recuperación de objetos. Está dividido en tres subpaquetes: AccesoDatos, que en general permite a la aplicación abstraerse del origen de los datos y de la lógica de su persistencia, logrando un bajo acoplamiento entre sus componentes; SQLite, que contiene las clases encargadas de acceder a la base de datos para manipular la persistencia de las entidades; y también se encuentra el subpaquete de ServiciosWeb, útil para obtener la información que la aplicación necesita para su funcionamiento.

Las clases de la capa de AccesoDatos acceden al paquete Entidades que contiene clases que no tienen comportamiento, sólo propiedades y son representaciones de entidades reales del dominio, siendo la mayoría clases persistentes.

Esta división por capas y paquetes obedece a la funcionalidad de las clases, las cuales han sido agrupadas de esta forma para lograr mayor desacoplamiento, reutilización y legibilidad de los diagramas. Como resultado de ello, se logra que la aplicación tenga un soporte más sencillo y una mayor flexibilidad, considerando la ventaja de poder cambiar un componente sin que sea necesario cambiar la aplicación, así como que poder agregarlos a la misma, aumentando las funcionalidades del sistema.

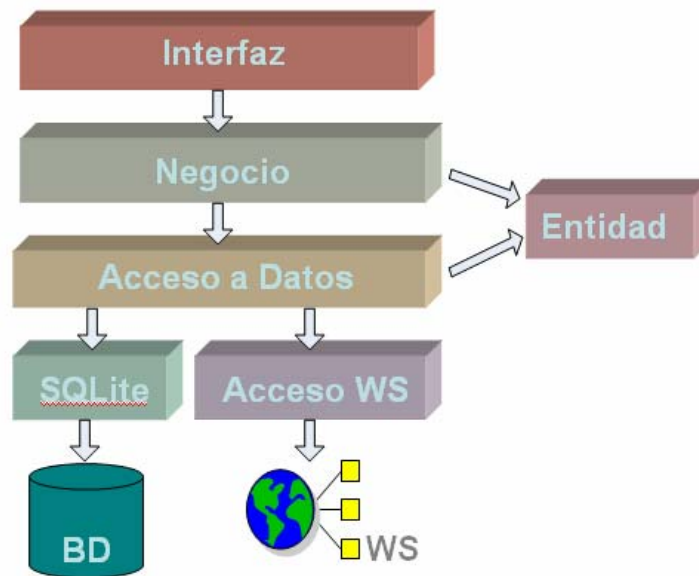


Figura 26. Arquitectura en capas.

3.6 Definiciones de diseño que se apliquen.

El diseño ha sido elaborado pensando en los usuarios finales, los cuales, no en todos los casos poseen suficientes conocimientos de computación, por tanto, se ha elaborado una interfaz amigable e intuitiva para lograr una mayor rapidez y factibilidad en el trabajo con la aplicación.

Los elementos que conforman la interfaz están distribuidos de forma organizada y elegante. Las ventanas contienen sólo la información necesaria, evitando sobrecargarlas.

Para diseñar la interfaz se tuvieron en cuenta los siguientes detalles:

- El usuario debe ver sólo las opciones que necesite en ese momento y no otras.
- Únicamente se puede ejecutar una acción a la vez.
- Para cambiar de acción el usuario debe decidir qué hacer con la que tiene en curso.
- En cada ventana se muestra la información necesaria.

3.6.1 Interfaz de usuario.

La interfaz es el “rostro” del sistema, lo cual tiene una gran importancia a la hora de hacer el diseño para la misma. Mantener la uniformidad es otro de los aspectos a tener en cuenta para mantener en equilibrio estilo y elegancia.

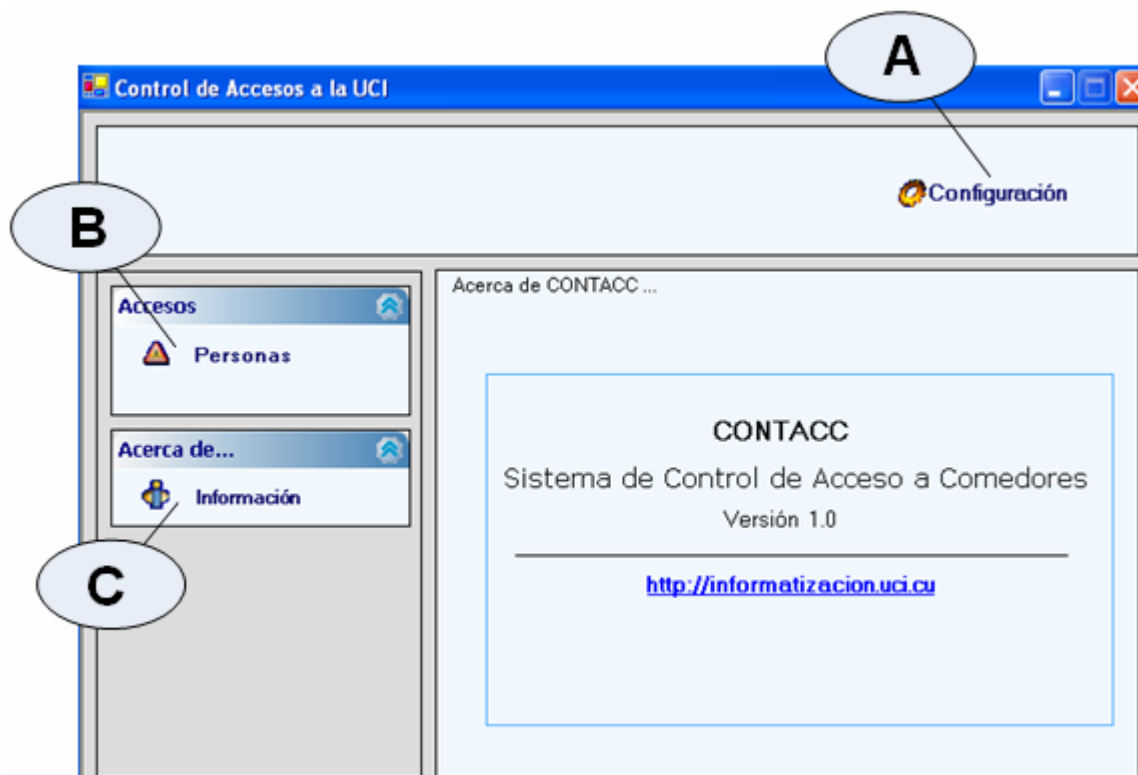


Figura 27. Interfaz de Usuario.

- A – Se accede a la ventana de configuración del sistema.
- B – Se accede a la ventana de Personas donde se registran los accesos y se muestran los datos de dichas personas.
- C – Se muestra la Información del sistema.



Figura 28. Personas.

- A – Se muestran los datos de las personas.
- B – Se muestra la foto de la persona.

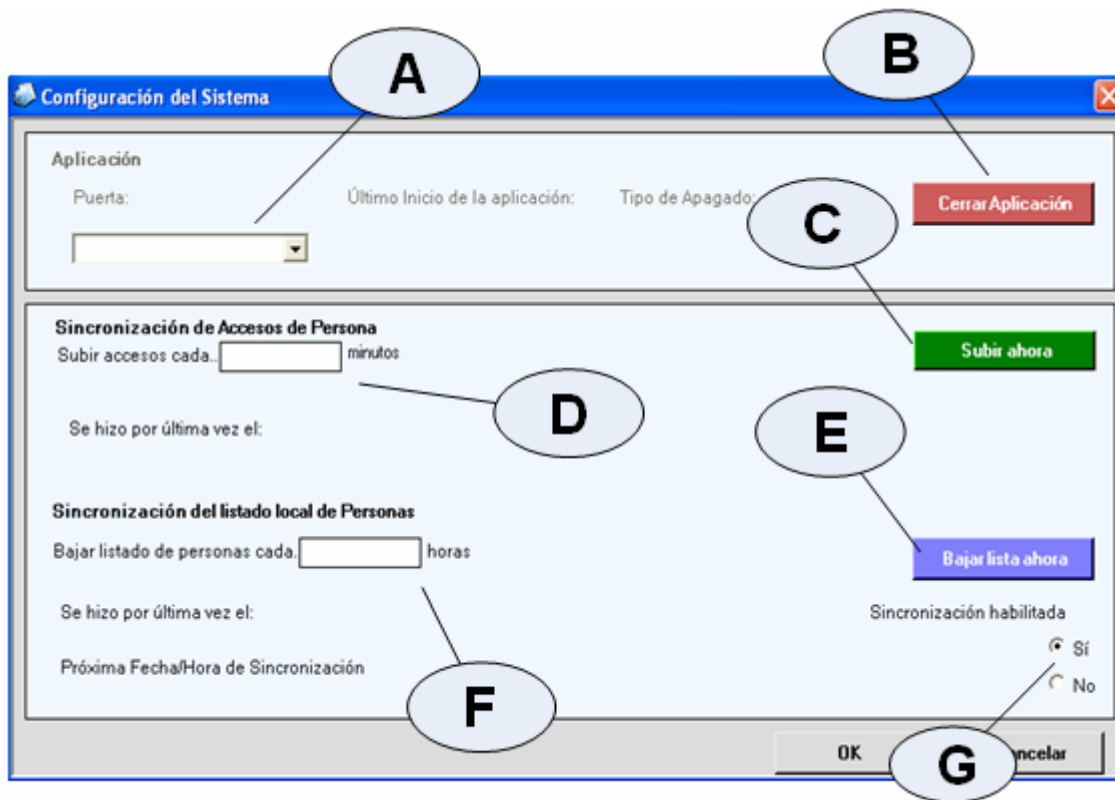


Figura 29. Configuración.

- A – Se muestra la puerta seleccionada de las que se encuentran inactivas.
- B – Botón que se utiliza para cerrar la aplicación.
- C – Botón que sincroniza los accesos de las personas.
- D – Se define cada qué tiempo se sincronizan los accesos de las personas.
- E – Botón que sincroniza los listados de las personas.
- F – Se define cada qué tiempo se sincronizan las personas.
- G – Habilitar o deshabilitar la sincronización.

3.7 Tratamiento de errores.

Con el objetivo de prevenir errores por parte del usuario, sólo se le brindan las opciones necesarias, a la hora de efectuar cualquier operación, por ejemplo, se deshabilitan ciertos botones si no tienen que ser utilizados en ese momento, y si se escriben datos incorrectos, se le muestra un mensaje de error.

Del lado del Servidor Web, se garantiza que los datos suministrados por los usuarios, se almacenen íntegros y no existan inconsistencias. Se verifican los campos obligatorios, y se revisa el tipo de dato. Los mensajes de error resaltan la atención al ser mostrados en color rojo. El usuario no necesita tener un elevado nivel cognoscitivo de la computación para entender las excepciones que podrían ocurrir en tiempo de corrida de la aplicación, pues se manipulan de forma tal que se genera un registro de errores en un fichero para una eventual revisión por parte del administrador.

3.8 Conclusiones.

En este capítulo se ha llevado a cabo la descripción de las clases y demás elementos que constituyen una guía excelente y confiable para la implementación. Se obtuvo además el propio diagrama de clases del análisis y el del diseño, se definieron, a partir del mismo, cuáles serían las clases persistentes, y luego, se construyó el modelo de datos y el diseño de la base de datos respectivamente.

También se definieron las propuestas para el diseño de la interfaz, teniéndose en cuenta lo mejor y más factible para el usuario final de la aplicación, así como los tratamientos de errores.

Por otra parte se llega a la conclusión de que es necesario que dicha aplicación cumpla con estándares de codificación y principios de diseño previamente estudiados y probados, que garanticen que cualquier programador pueda comprender con éxito cómo está implementado el sistema.

Capítulo 4. Implementación

4.1 Introducción.

En el presente capítulo se muestra el diagrama de despliegue y el de componentes, este primero define los nodos que conforman la estructura física de la aplicación, mientras que el de componentes representa la parte modular del sistema, o sea, contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.), ambos diagramas conforman lo que se conoce como modelo de implementación.

4.2 Diagrama de despliegue.

El sistema consta de un componente principal que es el Sistema Control de Acceso a Comedores en la UCI, CONTACC, ubicada en cada una de las puertas. Esta se conecta a un Servidor Web a través del protocolo HTTP donde se encuentra el Sistema de Gestión de Comensales, el cual hace uso de un Servidor de Base de Datos, que provee al mismo de información sobre las personas que acceden por dichas puertas.

En el caso de la aplicación cliente, usa además bases de datos locales para comprobar la existencia de las personas consultando la información, y para registrar los accesos de las personas si no hay conexión con el Servidor Web. También se conecta a un repositorio de fotos a través del protocolo TCP/IP para buscar las fotos de las personas.

4.2.1 Representación del diagrama de despliegue.

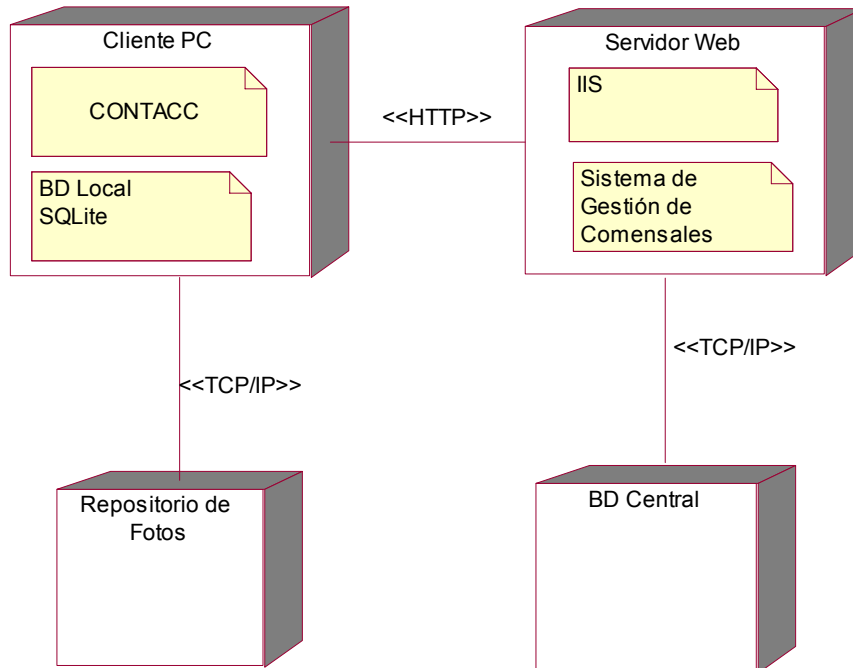


Figura 30. Diagrama de Despliegue.

4.3 Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

El diagrama de componentes del sistema propuesto consta de varios componentes, entre los más importantes se encuentran la interfaz de usuario, que le brinda al mismo todas las funcionalidades para interactuar con el sistema, el componente de negocio, en el cual se maneja toda la lógica de la aplicación, y el acceso a las capa de acceso a datos, el mismo maneja las transacciones ya sea a una base de datos local, mediante el componente SQLite o a una base de datos central a través del componente de Servicios Web.

4.3.1 Representación del diagrama de componentes.

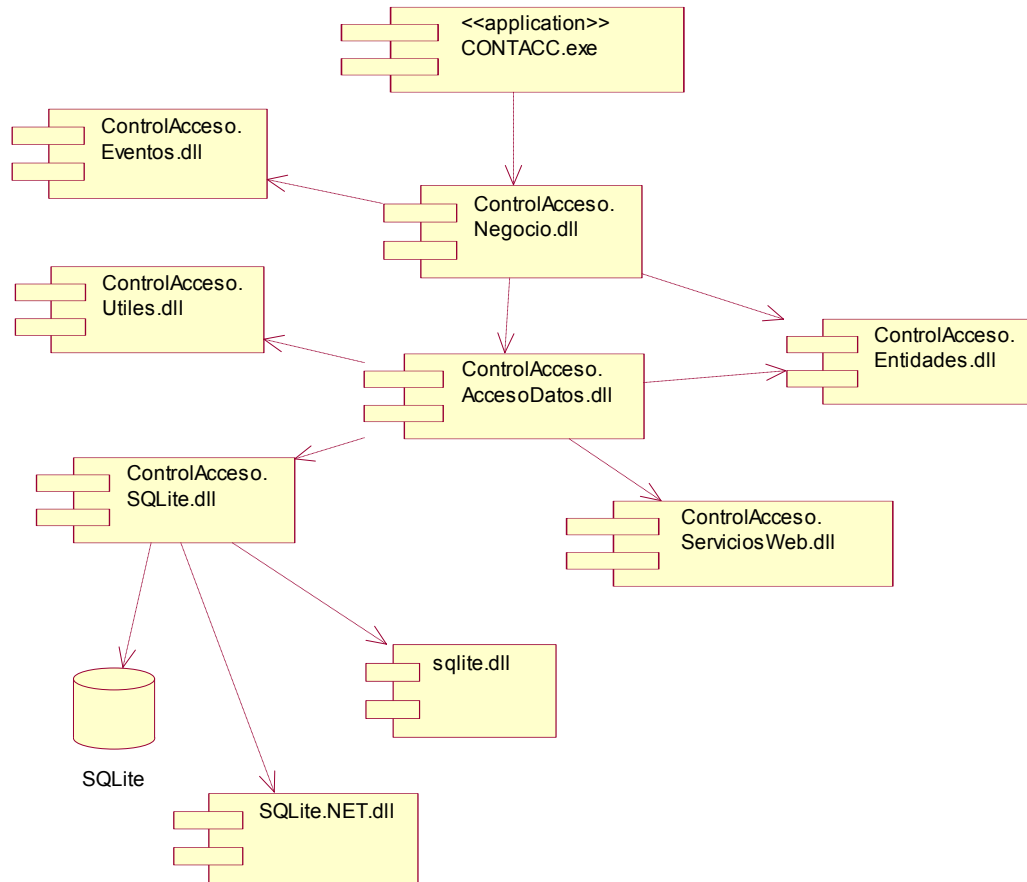


Figura 31. Diagrama de Componentes.

4.4 Conclusiones.

En este capítulo se planteó el modelo de implementación a partir del diagrama de componentes y el de despliegue, donde se definen los nodos y componentes que conforman la estructura física de la aplicación.

Capítulo 5. Estudio de Factibilidad

5.1 Introducción.

Para llevar a cabo el desarrollo de un proyecto de software es importante evaluar la factibilidad del mismo antes de su elaboración, pues es necesario determinar si es viable o no con la intención de minimizar los gastos en tiempo de planificación, esfuerzo y fondos, conociendo si es conveniente llevarlo a cabo. En este capítulo se hará un estudio del costo y los beneficios del sistema propuesto. A partir de este estudio se obtendrá el tiempo de desarrollo del proyecto, así como la cantidad de personas que intervendrán y el costo que tendrá el mismo.

5.2 Planificación basada en casos de uso.

Paso 1. Cálculo de los Puntos de Casos de Uso Desajustados.

El primer paso para la estimación mediante el análisis de Puntos de Casos de Uso consiste en el cálculo de los Puntos de Casos de Uso Sin Ajustar. Este valor se calcula a partir de la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Donde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

El **Factor de Peso de los Actores sin ajustar (UAW)** se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1		
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2		
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2	6
Total			2	6

Tabla 13. Factor de peso de los actores sin ajustar.

UUCW = Σ cant actores * Peso

UAW=3*2=6 Factor de peso de los actores sin ajustar

El **Factor de Peso de los Casos de Uso Sin Ajustar (UUCW)** se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. Dicha complejidad se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo.

Tipo de CU	Descripción	Factor de peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	8	40
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15		
Total			8	40

Tabla 14. Factor de peso de los casos de uso sin ajustar.

UUCW = Σ cant CU * Peso

UUCW = $8 * 5 = 40$ Factor de peso de los CU sin ajustar

UUCP = **UAW + UUCW**

UUCP = $6 + 40 = 46$ Puntos de casos de uso sin ajustar

Paso 2. Cálculo de los Puntos de Casos de Uso Ajustados.

UCP = UUCP * TCF * EF

Donde:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se determina mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema.

Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy relevante.

Factor	Descripción	Peso	Comentarios	Valor Asignado	Total
T1	Sistema distribuido	2	El sistema será utilizado desde varias máquinas a la vez.	5	10
T2	Tiempo de respuesta	1	La velocidad es limitada por la conectividad.	3	3
T3	Eficiencia del usuario final	1	Requiere de conocimiento para tratar con la tecnología y el sistema.	2	2
T4	Funcionamiento Interno complejo	1	No existen cálculos complejos.	0	0
T5	El código debe ser reutilizable	1	El código será reutilizable.	5	5
T6	Facilidad de instalación	0.5	No implica una instalación compleja.	3	1.5
T7	Facilidad de uso	0.5	Fácil de usar.	5	2.5
T8	Portabilidad	2	No se requiere que el sistema sea portable.	0	0
T9	Facilidad de cambio	1	Costo moderado de mantenimiento.	4	4
T10	Concurrencia	1	Existe concurrencia.	3	3
T11	Incluye objetivos especiales de seguridad	1	Seguridad normal.	3	3
T12	Provee acceso directo a terceras partes	1	Si los tiene.	4	4
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	Presenta facilidad de uso pero por su importancia se adiestrarán.	3	3
Total				37	41

Tabla 15. Factor de complejidad técnica.

$$\text{TCF} = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i)$$

(Donde Valor es un número del 0 al 5)

$$\text{TCF} = 0.6 + 0.01 * 41 = 1.01 \text{ Factor de complejidad técnica}$$

El factor de ambiente (EF) está relacionado con las habilidades y el entrenamiento del grupo involucrado en el desarrollo del sistema. Estos factores son los que se contemplan en el cálculo del **Factor de ambiente (EF)**. El cálculo del mismo se determina mediante la cuantificación de un conjunto de factores con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy relevante.

Factor	Descripción	Peso	Comentarios	Valor Asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	El grupo no está lo suficientemente familiarizado con el modelo.	3	4.5
E2	Experiencia en la aplicación	0.5	La mayoría del grupo ha trabajado durante algún tiempo en ésta aplicación.	3	1.5
E3	Experiencia en la orientación a objetivos.	1	La mayoría del grupo programa orientado a objetos.	4	4
E4	Capacidad del analista líder.	0.5	El líder es un Ing. Informático.	5	2.5
E5	Motivación.	1	Existe motivación por parte del grupo.	3	3
E6	Estabilidad de requerimientos	2	Existe probabilidad de cambios.	4	8
E7	Personal Part-Time	-1	No todo el grupo trabaja a tiempo completo.	2	-2
E8	Dificultad del lenguaje de programación	-1	Se usará lenguaje C#.	4	-4
Total				28	17.5

Tabla 16. Factor de ambiente.

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$EF = 1.4 - 0.03 * 17.5 = 0.875 \text{ Factor de ambiente.}$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 46 * 1.01 * 0.875 = 40.6525 \text{ Puntos de casos de uso ajustados}$$

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso.

El esfuerzo en horas hombres se calcula aplicando la siguiente ecuación matemática:

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para calcular CF

CF = 20 horas-hombre (si $Total_{EF} \leq 2$)

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

$Total_{EF} = Cant\ EF < 3$ (entre E1 –E6) + $Cant\ EF > 3$ (entre E7, E8)

Como $Total_{EF} = 2 + 0$

$Total_{EF} = 2$

CF = 20 horas-hombre (porque $Total_{EF} \leq 2$)

Se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso.

$E = 40.6525 * 20 = 813.05$ Horas-Hombre

Paso 4. Calcular esfuerzo de todo el proyecto.

Finalmente, para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación: (CRUZ, T. L. O. y P. P. ZURITA, 2006)

Actividad	% esfuerzo	Valor esfuerzo (horas-hombre)
Análisis	10%	203.2625
Diseño	20%	406.525
Implementación	40%	813.05
Prueba	15%	304.89375
Sobrecarga	15%	304.89375
Total	100%	2032.625

Tabla 17. Esfuerzo del proyecto.

Si el esfuerzo total (E_T) es de 2032.625 **horas-hombre** y por cada 240 horas laborables tenemos 1 mes, eso daría un esfuerzo total de 8.46927083 **mes-hombre**.

Esto quiere decir que 1 persona puede realizar el sistema analizado en aproximadamente 8 meses.

Costo del Proyecto

Se asume como salario promedio mensual \$100.00

Cantidad de hombres: CH = 2

Costo Hombre-Mes: CHM = 2 * Salario Promedio

CHM = 200.00 \$/mes

Costo = CHM * E_T / CH

Costo = 200.00 * 8.46927083 = 1693.85417

Costo = \$1693.85417 \approx \$1694

Tiempo \approx 8 meses

Tiempo \approx 4 Para dos hombres

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto, el mismo pudiera desarrollarse en 4.1 meses y su costo total se estima que sea de \$1644.

5.3 Beneficios tangibles e intangibles.

El Sistema de Control de Acceso a Comedores en la UCI no es un software con fines comerciales, pues está orientado a resolver los problemas existentes en la Universidad, aunque puede ampliarse para convertirlo en una solución general, capaz de aplicarse a cualquier empresa o institución.

Su principal objetivo es ayudar al control de acceso de las personas a los comedores, por tanto, los beneficios inmediatos son mayormente intangibles:

- Ahorro de tiempo en la gestión de acceso al comedor.
- Posibilidad de ver reportes en los que se visualicen la cantidad de comensales por tipo accedidos hasta el momento.
- Aumento de la organización interna.

De forma directa también se obtendrán beneficios tangibles ya que una vez desplegado el sistema se podrán controlar con mayor facilidad los accesos de las personas a los comedores, siendo así posible una mejor planificación de la cantidad de alimentos a elaborar, lo que propicia menores gastos de recursos.

5.4 Análisis de costos y beneficios.

El gasto de este sistema radica mayormente en los lectores de código de barras necesarios para controlar el acceso al comedor, pero dicho gasto se soporta en las ventajas que brindará el sistema, al controlar de manera segura y eficiente la gestión del acceso de las personas a los comedores.

Se investigaron otras soluciones de control de acceso disponibles en el mundo, pero ninguna de ellas se ajusta a las características del entorno de la Universidad, y además son demasiado caras.

El sistema puede ser extendido para uso general, convirtiéndose en fuente de ingresos como un producto comercializable.

5.5 Conclusiones.

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser puesto en marcha. Los resultados de dicho análisis dieron como resultado la aceptación de la propuesta.

Conclusiones Generales

Como resultado de la investigación llevada a cabo para el desarrollo de este documento y del propio sistema que se describe en su contenido, es que surge el sistema “CONTACC”, propuesta que apoya el proceso de gestión de control de acceso a comedores en la UCI. Esta aplicación responde al objetivo principal propuesto inicialmente y además específicamente logra registrar la entrada a los comedores de las personas vinculadas a la UCI, garantizando que esto tenga lugar una sola vez y por la puerta correspondiente, actualizar la información de las personas que pasan por cada puerta, brindar reportes sobre las personas que fueron teniendo acceso y hacer más seguro y confiable el proceso de identificar a dichas personas.

El sistema es considerado de fácil uso para los usuarios finales de la aplicación, pues su interfaz es amigable, además es flexible, lo cual le permite adaptarse con facilidad a variaciones en la estructura del flujo informativo, además como ventaja fundamental está el hecho de que permite configurar los tiempos de sincronización de forma manual o automática cada determinado momento, ideal para las necesidades actuales de la Universidad.

Su arquitectura está diseñada teniendo en cuenta la integración con otras aplicaciones, lo cual asegura independencia entre la parte visual de la aplicación, su negocio, su acceso a datos y los datos que utiliza, permitiendo la posibilidad de realizar cambios en algunas de estas capas sin necesidad de actualizar el sistema completo.

Para el desarrollo del trabajo fue necesario utilizar el Gestor de Bases de Datos SQLite para las bases de datos locales y un Servicio Web para el complemento de la información que se maneja en el sistema, el lenguaje de programación utilizado fue C Sharp, y la plataforma .Net para la implementación de la aplicación. Para la modelación del sistema se empleó RUP como metodología de desarrollo y UML como lenguaje de modelado.

Recomendaciones

De forma general fueron cumplidos los objetivos propuestos al inicio de este trabajo, no obstante, durante su desarrollo, han ido aflorando una serie de ideas y recomendaciones que podrían tenerse en cuenta para futuras versiones de la solución, de manera que pueda obtenerse una aplicación más óptima.

Para ello se recomienda:

- Extender y adaptar el sistema de manera que pueda ser utilizado no sólo en la Universidad, sino en cualquier entorno, en el que se requiera de controlar accesos de las personas vinculadas al mismo.
- Migrar el sistema a Software Libre.
- Publicar o dar a conocer la forma en que la aplicación CONTACC utiliza SQLite, ya que muchas aplicaciones en la UCI pudieran utilizarla para guardar datos locales.

Además se recomienda dar mantenimiento y actualización periódica a la aplicación, logrando así la fiabilidad y funcionamiento óptimo de la misma y de la información que se gestiona a través de ella.

Referencias Bibliográficas

BOOCH, G., RUMBAUGH, J., JACOBSON, I. *El Proceso Unificado de Desarrollo de Software*. La Habana, Félix Varela, 2004. 256 p.

CRUZ, T. L. O. y P. P. ZURITA. *Sistema de Control de Acceso a la Universidad de las Ciencias Informáticas*. Ciudad de La Habana, Instituto Superior Politécnico “José Antonio Echeverría”, Universidad de las Ciencias Informáticas, Junio 2006. 96. p.

IDENTIFIC-AR. *Lector de código de barras láser* 2006. [Disponible en: <http://identific-ar.com.ar/ms9520.htm>]

MARIA, Y. *Envases y Embalajes - Códigos de barras*, 2005. [Disponible en: <http://www.monografias.com/trabajos11/yantucod/yantucod.shtml#acerca>]

MARTÍN, M. A. G. *Sistema de Control de Accesos*. Ciudad de La Habana Instituto Superior Politécnico “José Antonio Echeverría”, Universidad de las Ciencias Informáticas, Junio 2005. 116. p.

MEDINA, D. D. A. *Sistema de Gestión Académica: Módulo de Diseño y Distribución de Reportes*. Ciudad de La Habana, Instituto Superior Politécnico “José Antonio Echeverría”, Universidad de las Ciencias Informáticas, Junio 2005. 133. p.

Bibliografía

El Modelo COCOMO. Disponible en: <http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm>

Tecnología .NET. 02/08/2004. [Disponible en: <http://globaliza.blogia.com/temas/tecnologia.net.php>

AMTEL. *AMWATCH Sistemas de Control de Acceso*, 2005. [Disponible en: <http://www.amtel-security.com/spanish/products/ACS.html>

CHRISTENSEN, E.; F. CURBERA, *et al.* *Lenguaje de descripción de Servicios Web (WSDL)*, 13 Enero 2003. [Disponible en: <http://www.microsoft.com/spanish/msdn/articulos/archivo/090201/voices/wsdl.asp>

CORPORATION, M. *Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios*, 31 Mayo 2004. [Disponible en: <http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp>

CORPORATION, M. *Características futuras del lenguaje de programación C#*, 5 Mayo 2003. [Disponible en: <http://www.microsoft.com/spanish/msdn/articulos/archivo/050503/voices/vbconC.asp>

CRUZ, A. D. V. *Common Language Runtime*, 02/Mayo/2005. [Disponible en: http://www.elguille.info/colabora/NET2005/vecrado_CommonLanguageRuntime.htm

CRUZ, T. L. O. and P. P. ZURITA. *Sistema de Control de Acceso a la Universidad de las Ciencias Informáticas*. Ciudad de La Habana, Instituto Superior Politécnico “José Antonio Echeverría”, Universidad de las Ciencias Informáticas, junio 2006. 96. p.

DESARROLLOWEB.COM. *Qué es SQL*, 2005. [Disponible en: <http://www.desarrolloweb.com/articulos/262.php>

IBIX. *Tecnologías de Información*, 30 Septiembre 2006. [Disponible en: <http://www.ibix.com.mx/HandPunch.htm>

IBIX. *Tecnologías de Información*, 30 Septiembre 2006. [Disponible en:
http://www.ibix.com.mx/BioControl_1000_A11Prox.htm

IBIX. *Tecnologías de Información*, 30 Septiembre 2006. [Disponible en:
http://www.ibix.com.mx/BioControl_1000.htm

IBIX. *Tecnologías de Información*, 07 Junio 2007. [Disponible en:
<http://www.ibix.com.mx/UCtrlEthernetRelojChecador.htm>

IBIX. *Tecnologías de Información*, 20 Junio 2006. [Disponible en:
http://www.ibix.com.mx/Sistemas_Biometricos.htm

JACOBSON, I.; G. BOOCH, *et al. Proceso unificado de desarrollo de software*. La Habana Félix Varela 2004. p.

LARMAN, C. *UML y patrones*. La Habana Félix Varela 2004. 921 p.

PANIAGUA, A. B. *Tutorial de XML*, 05/12/1999. [Disponible en:
<http://www.dat.etsit.upm.es/~abarbero/curso/xml/xmltutorial.html>

PÉREZ, J. S. *.NET Framework, uno para todos y todos para uno*, 2007. [Disponible en:
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art65.asp>

SOMOSLIBRES.ORG. *¿Qué es SQLite?*, 2007. [Disponible en:
<http://www.softwarelibre.org.pe/modules.php?name=News&file=article&sid=183>

UTN. *QuickPass: un sistema de control de accesos y presentismo*, 2007. [Disponible en:
http://www.sceu.frba.utn.edu.ar/e-learning/index.php?option=com_content&task=view&id=50&Itemid=15

WIKIPEDIA. *C#*, 14 junio 2007. [Disponible en: http://es.wikipedia.org/wiki/C_Sostenido

BIBLIOGRAFÍA

WIKIPEDIA. *COCOMO Básico*, 14 marzo 2007. [Disponible en:
http://es.wikipedia.org/wiki/COCOMO_B%C3%A1sico

WIKIPEDIA. *Microsoft Visual Studio .Net*, 31 may 2007. [Disponible en:
http://es.wikipedia.org/wiki/MS_Visual_Studio_.NET

WIKIPEDIA. *NET*, 14 jun 2007. [Disponible en: <http://es.wikipedia.org/wiki/.NET>

WIKIPEDIA. *Servicio Web*, 19 junio 2007. [Disponible en: http://es.wikipedia.org/wiki/Servicio_Web

WIKIPEDIA. *SQLite*, 20 marzo 2007. [Disponible en: <http://es.wikipedia.org/wiki/SQLite>

WIKIPEDIA. *WSDL*, 27 febrero 2007. [Disponible en: <http://es.wikipedia.org/wiki/WSDL>

WIKIPEDIA. *XML*, 19 junio 2007. [Disponible en: <http://es.wikipedia.org/wiki/XML>

Anexos

Anexo 1.

Descripción del caso de uso del negocio, Diagrama de Actividad y Modelo de Objetos del caso de uso crítico Acceder al comedor.

Caso de uso del negocio: Acceder al comedor	
Actores del negocio: Persona (Inicia)	
Propósito: Permitir y registrar los accesos de las personas a los comedores.	
Resumen: El caso de uso lo inicia la persona cuando se presenta en la puerta de entrada al comedor, este debe presentar su solapín para que se valide su acceso y pase, en caso de existir problemas se denegará el acceso y se retirará.	
Curso normal de los eventos	
Acción del actor	Respuesta del negocio
1. La persona se presenta en la puerta del comedor.	2. El cajero solicita su solapín o documento firmado.
3. La persona muestra su identificación.	4. El cajero chequea la credencial y verifica que exista y tenga acceso. En caso de que esté todo correcto, pasa al paso 6. En caso contrario, le informa a la persona que hay un error y que su acceso no podrá ser gestionado mientras no se resuelva el problema.
5. La persona se retira del comedor.	
	6. El cajero informa a la persona que tiene acceso.
7. La persona entra al comedor.	8. Se registra acceso válido.
Prioridad: Responde al principal objetivo de automatización al resolver gran parte de los problemas actuales.	

Tabla 18. Diagrama de actividad “Acceder al comedor”.

Estos diagramas permiten ver el flujo de actividades que se lleva a cabo, en ellos se identifican las actividades objeto de automatización. En este caso las actividades objeto de automatización están señaladas en el diagrama.

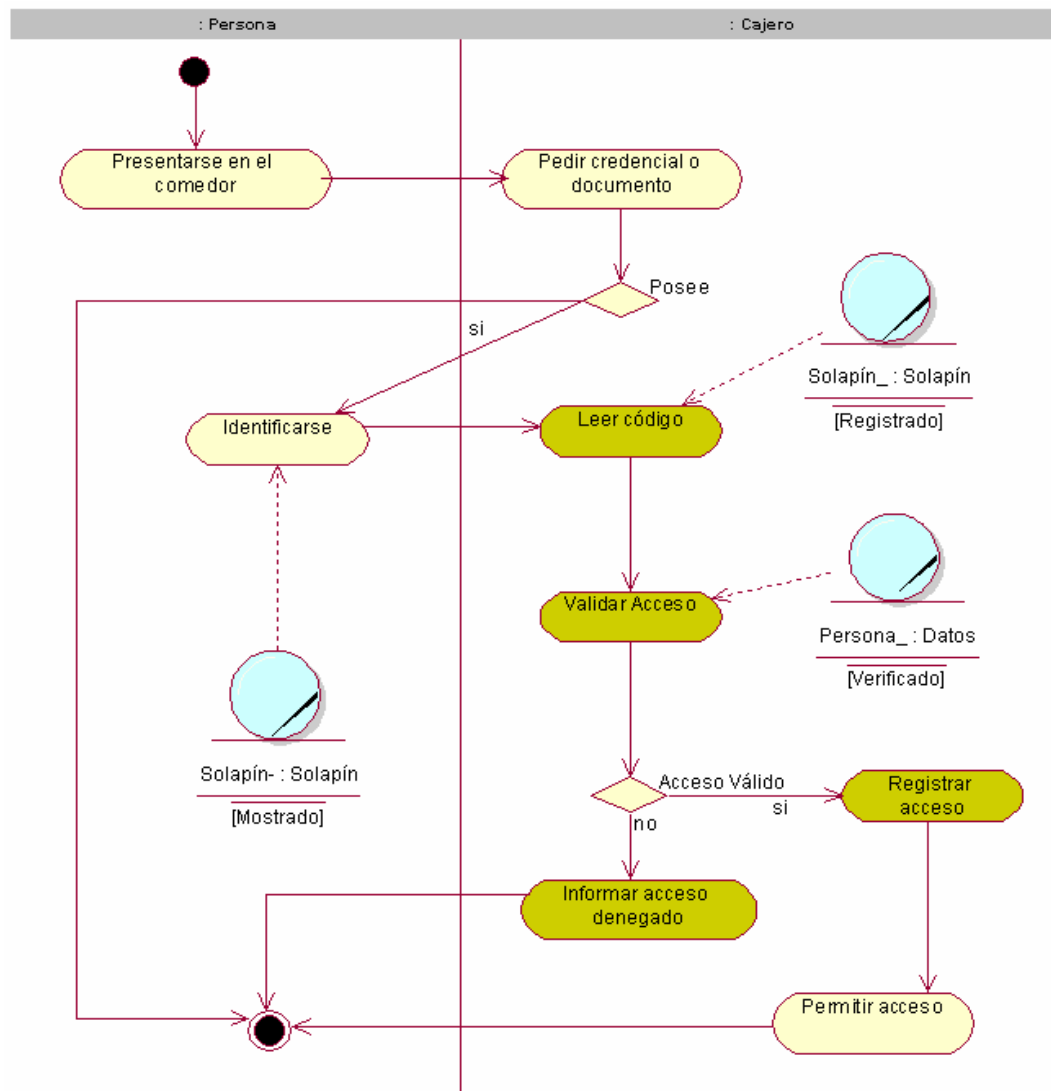


Figura 32. Diagrama de actividad “Acceder al comedor”.

En este diagrama se representan los trabajadores del negocio relacionados con las entidades. En este caso solamente existen dos trabajadores del negocio.

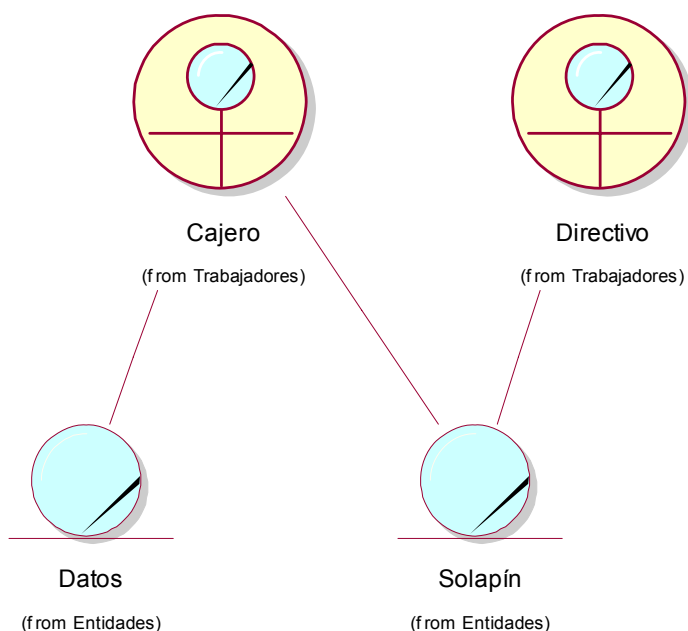


Figura 33. Modelo de Objeto.

Anexo 2.

Descripción de los Diagramas de caso de uso.

Se presenta una descripción detallada de los casos de uso donde se expone: nombre del caso de uso, propósito del caso de uso en cuestión, actores que lo inicializan, un pequeño resumen, las referencias, las acciones del actor y la respuesta del sistema así como los flujos alternativos que puede tomar el caso de uso.

Nombre del caso de uso:	Registrar Acceso
Actores:	Cajero “(inicia)”
Propósito:	Archivar el acceso de todas las personas que han accedido.
Resumen:	El caso de uso se inicia cuando el cajero recibe al comensal en la puerta del comedor, se lee el código de barra que contiene la credencial con un lector de código de barras o introduciendo los datos manualmente en la aplicación, luego el sistema registra aquellos accesos que son válidos, o sea, los correspondientes a las personas autorizadas por esa puerta y que no hayan accedido con anterioridad; y si ocurre algún error, el cajero denegará el acceso según la información mostrada por el sistema.
Referencias:	RF2, RF2.1, RF2.2
Precondiciones:	Para registrar el acceso el cajero debe utilizar el lector de código de

	barra o introducir manualmente el identificador de la persona para verificar su validez.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1- El cajero utiliza el lector de código de barra o introduce manualmente el identificador de la persona.	1.1- El sistema verifica si la persona existe. 1.2- Si la persona existe, verifica si ha accedido con anterioridad. 1.3- En caso de no haber accedido, se permite el acceso. 1.4- El sistema registra el acceso válido. 1.5- El sistema muestra la tabla descriptiva con la cantidad de comensales accedidos hasta el momento.
2- El cajero permite la entrada del usuario al comedor y finaliza el proceso.	
Curso Alterno (1.2)	
	1.2- En caso de no existir persona, el sistema envía un mensaje de aviso.
2- El cajero deniega la entrada del usuario al comedor y finaliza el proceso.	
Curso Alterno (1.3)	
	1.3- En caso de haber accedido, el sistema envía un mensaje de aviso.
2- El cajero deniega la entrada del usuario al comedor y finaliza el proceso.	
Prioridad	Crítico

Tabla 19. “ Registrar_Acceso”.

Nombre del caso de uso:	Sincronizar Persona
Actores:	Sincronizador “(inicia)”
Propósito:	Permitir actualización.
Resumen:	Después de activada la aplicación, tiene lugar la sincronización de forma manual por el administrador del sistema, o automática, siendo posible predeterminedar el momento de hacerlo. En este caso de uso se intercambia información entre la aplicación y el Servidor Web, actualizándose el listado de las personas de la base de datos local con el propio Servidor Web.
Referencias:	RF3.2
Precondiciones:	Para sincronizar las personas, es necesario que exista conexión.
Poscondiciones:	Actualización de las personas.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema

1- El sincronizador ejecuta la actualización.	1.1- El sistema se conecta con el Servidor Web. 1.2- Se intercambia información. 1.3- Se actualizan las personas.
Prioridad	Crítico

Tabla 20. "Sincronizar_Persona".

Nombre del caso de uso:	Sincronizar Acceso
Actores:	Sincronizador "(inicia)"
Propósito:	Permitir actualización.
Resumen: Después de activada la aplicación, tiene lugar la sincronización de forma manual por el administrador del sistema, o automática, siendo posible predeterminedar el momento de hacerlo. En este caso de uso se intercambia información entre la aplicación y el Servidor Web, actualizándose remotamente en el Servidor Web los accesos válidos que fueron registrados en la base de datos local.	
Referencias:	RF3.2
Precondiciones:	Para sincronizar los accesos de las personas, es necesario que exista conexión.
Poscondiciones:	Actualización de los accesos de las personas.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1- El sincronizador ejecuta la actualización.	1.1- El sistema se conecta con el Servidor Web. 1.2- Se intercambia información. 1.3- Se actualizan los accesos de las personas.
Prioridad	Crítico

Tabla 21. "Sincronizar_Acceso"

Nombre del caso de uso:	Mostrar Datos
Actores:	Cajero "(inicia)"
Propósito:	Mostrar los datos de los comensales.
Resumen: El caso de uso se inicia cuando el cajero recibe al comensal en la puerta del comedor, se lee el código de barras que contiene la credencial con un lector de código de barras o introduciendo los datos manualmente en la aplicación. En el sistema se muestra, de la persona existente, datos como nombre, primer y segundo apellidos, número de solapín, foto y tipo de persona; para el estudiante se muestra facultad, grupo y año, y para cualquier otro tipo se muestra el área a la cual corresponde.	
Referencias:	RF4, RF4.1, RF4.2
Precondiciones:	Para mostrar los datos de las personas es necesario que la persona exista.
Poscondiciones:	Mostrar los datos.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema

1- El cajero utiliza el lector de código de barras o introduce manualmente el identificador de la persona.	1.1- El sistema verifica si la persona existe.
	1.2- El sistema muestra los datos correspondientes al tipo de persona.
Curso Alterno	
	1.1- En caso de no existir persona, el sistema envía un mensaje de aviso.
Prioridad	Crítico

Tabla 22. “Mostrar_Datos”.

Nombre del caso de uso:	Mostrar Reportes
Actores:	Cajero “(inicia)”
Propósito:	Mostrar los reportes.
Resumen: El caso de uso se inicia cuando el cajero recibe al comensal en la puerta del comedor, se lee el código de barras que contiene la credencial con un lector de código de barras o introduciendo los datos manualmente en la aplicación. En el sistema se muestra un reporte temporal del número de comensales por tipo accedidos en la sesión de alimentación, incrementándose en la medida en que se registran.	
Referencias:	RF5
Precondiciones:	Para mostrar los reportes es necesario que se haya registrado un acceso válido.
Poscondiciones:	Se incrementa por cada acceso válido el comensal accedido por tipo.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1- El cajero utiliza el lector de código de barras o introduce manualmente el identificador de la persona.	1.1- El sistema muestra la tabla de reportes.
Prioridad	Crítico

Tabla 23. “Mostrar_Reportes”

Nombre del caso de uso:	Autenticar
Actores:	Usuario “(inicia)”
Propósito:	Permitir autenticarse.
Resumen: El caso de uso se inicia cuando el usuario administrador introduce el user y password para poder acceder a la configuración del sistema con el objetivo de realizar alguna acción sobre el mismo.	
Referencias:	RF1, RF1.1
Precondiciones:	El usuario debe estar registrado en la base de datos.
Poscondiciones:	Se habilitan las funcionalidades para la configuración.

Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario introduce user y password.	1.1- El sistema encripta la contraseña. 1.2- Busca el usuario y compara la contraseña. 1.3- Si son correctos se le asignan los permisos.
Curso Alterno	
	1.3 En caso de error se envía un mensaje de aviso.
Prioridad	Crítico

Tabla 24. "Autenticar".

Nombre del caso de uso:	Mostrar_Datos_Configuración
Actores:	Administrador del Sistema "(inicia)"
Propósito:	Mostrar los datos referentes a la configuración del sistema.
Resumen:	El caso de uso se inicia cuando el administrador del sistema accede a la configuración y observa una ventana que muestra datos de la configuración del sistema tales como: el último apagado de la aplicación, la última fecha de sincronización de las personas o del acceso de las personas, el último acceso y cómo fue apagada la aplicación.
Referencias:	RF3, RF3.3
Precondiciones:	El usuario debe estar autenticado como administrador del sistema.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1- El administrador accede a la configuración.	1.1- El sistema muestra datos de la configuración del sistema tales como: el último apagado de la aplicación, la última fecha de sincronización de las personas o del acceso de las personas, próxima fecha/hora de sincronización y el tipo de apagado de la aplicación.
Prioridad	Crítico

Tabla 25. "Mostrar_Datos_Configuración"

Nombre del caso de uso:	Actualizar_Datos_Configuración
Actores:	Administrador del Sistema "(inicia)"
Propósito:	Actualizar los datos en la configuración del sistema.
Resumen:	El caso de uso se inicia cuando el administrador del sistema accede a la configuración y observa una ventana que muestra los datos de la configuración, este puede actualizar o cambiar parámetros de configuración tales como: tiempo de sincronización para las personas o para los accesos de dichas personas y además puede actualizar la puerta correspondiente.
Referencias:	RF3.1

Precondiciones:	El usuario debe estar autenticado como administrador del sistema.
Poscondiciones:	Se actualizan los datos de la configuración.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1- El administrador accede a la configuración.	1.1- El sistema muestra las opciones: tiempo de sincronización de personas y de los accesos de personas, además asignar número de puerta.
2- El administrador selecciona una de las opciones y la modifica.	2.1- El sistema actualiza los datos modificados.
Sección "Tiempo de sincronización del listado local de personas"	
2- El administrador selecciona tiempo de sincronización de personas y lo modifica.	2.1- El sistema actualiza el tiempo de sincronización de las personas.
Sección "Tiempo de sincronización de Accesos de persona"	
2- El administrador selecciona tiempo de sincronización de accesos de personas y lo modifica.	2.1- El sistema actualiza el tiempo de sincronización de accesos de las personas.
Sección "Puerta"	
2- El administrador selecciona Puerta y la modifica.	2.1- El sistema asigna el número de puerta introducido.
Prioridad	Crítico

Tabla 26. "Actualizar_Datos_Configuración".

Anexo 3.

Diagramas de interacción.

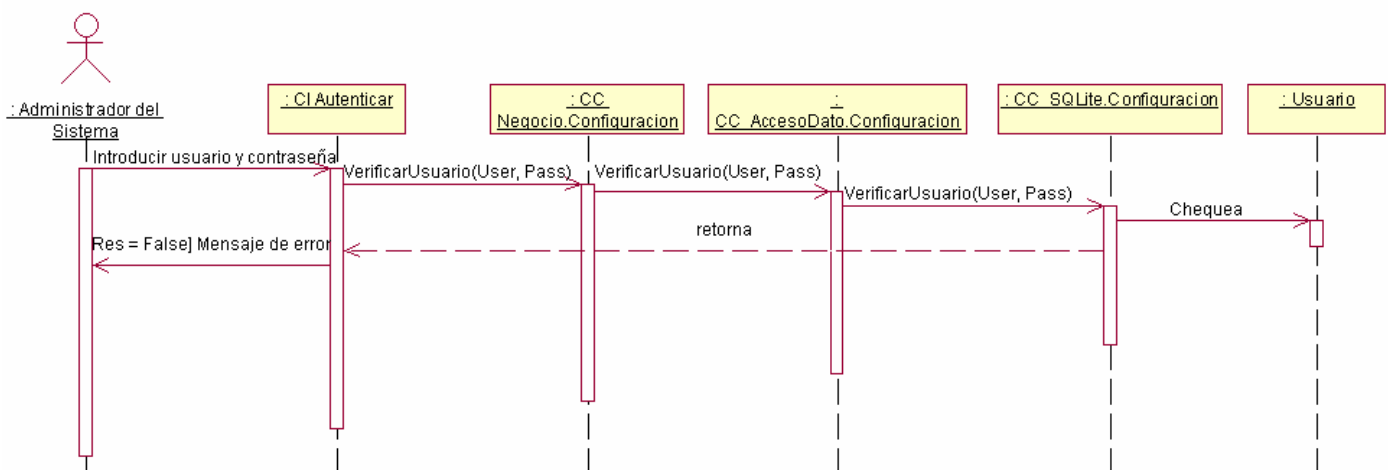


Figura 34. Diagrama de Secuencia del Caso de Uso “Autenticar”.

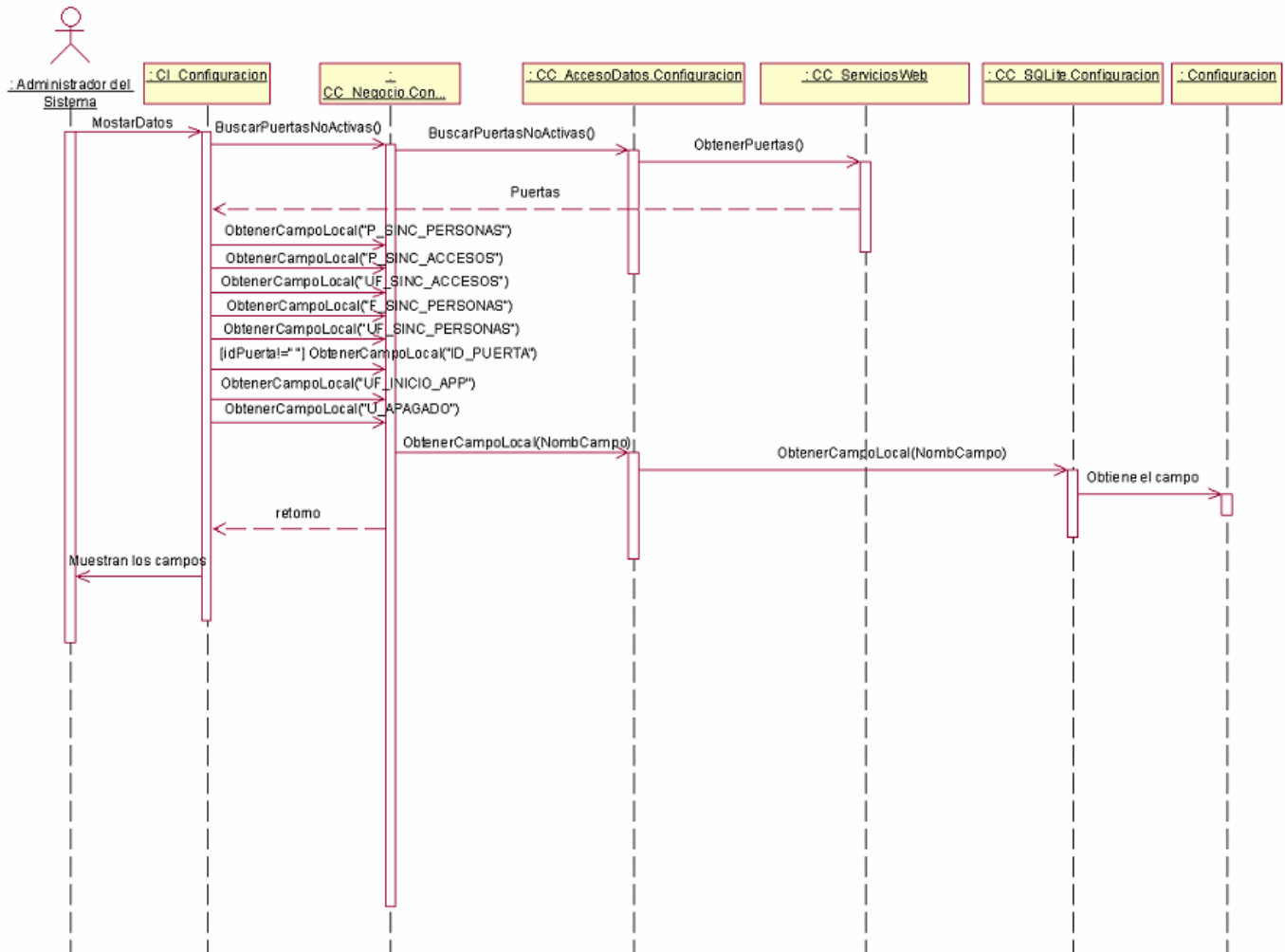


Figura 35. Diagrama de Secuencia del Caso de Uso “Mostrar_Datos_Configuración”.

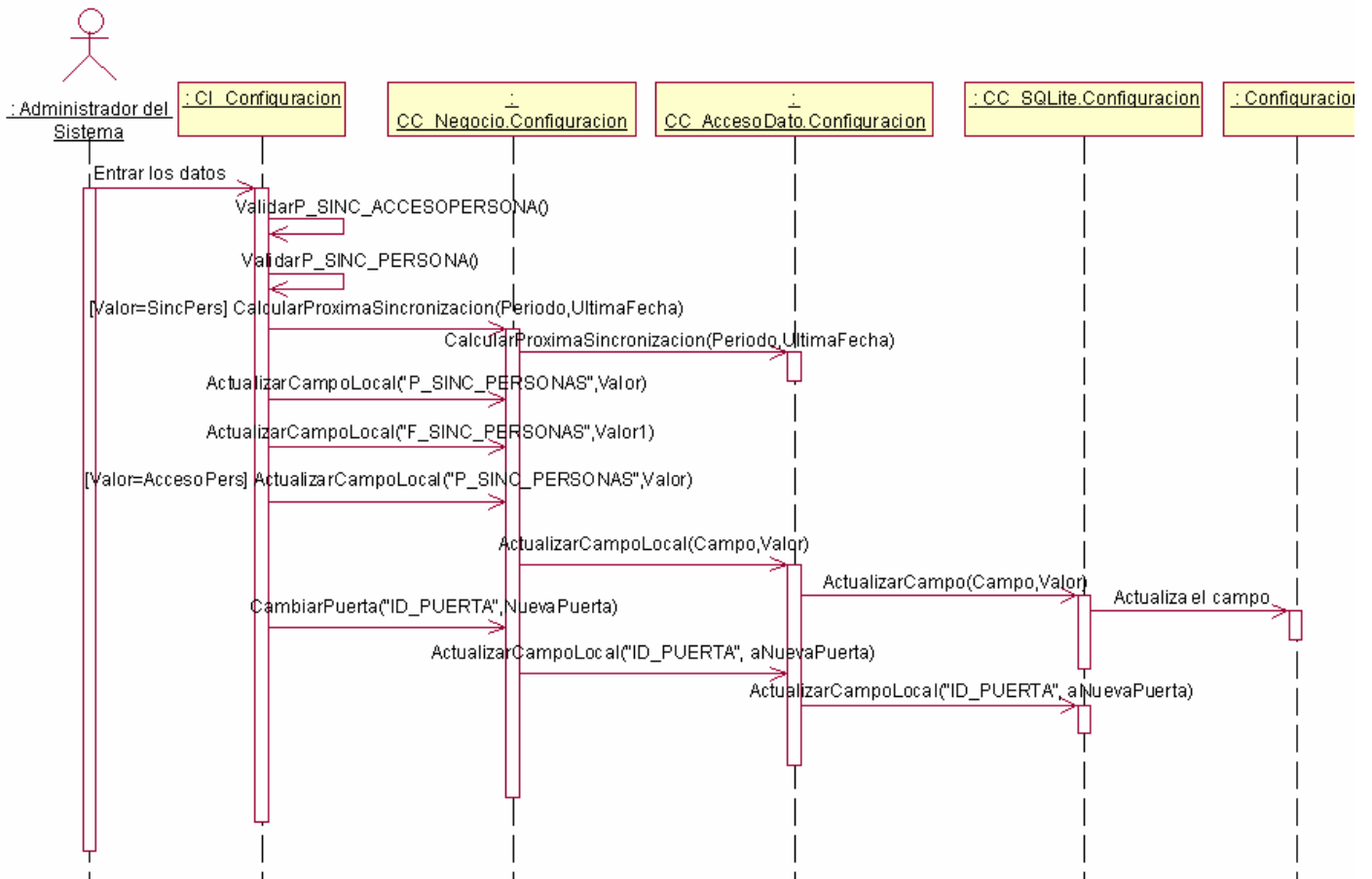


Figura 36. Diagrama de Secuencia del Caso de Uso “Actualizar_Datos_Configuración”.

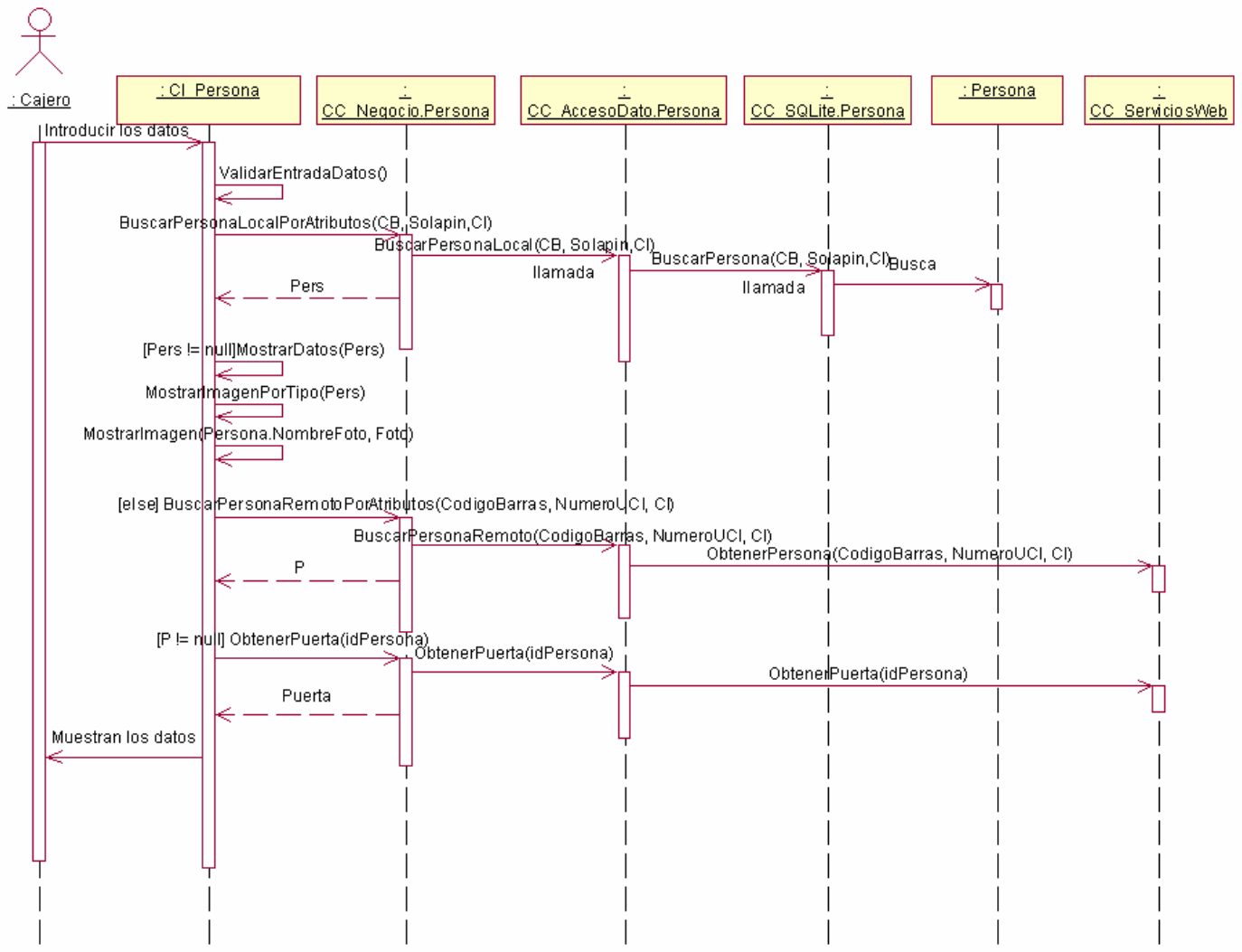


Figura 37. Diagrama de Secuencia del Caso de Uso “Mostrar_Datos”.

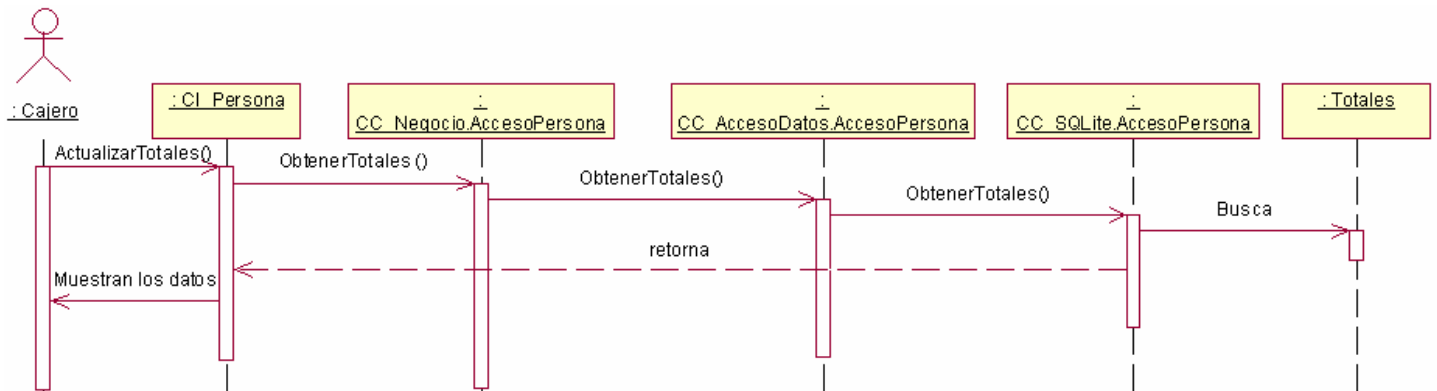


Figura 38. Diagrama de Secuencia del Caso de Uso “Mostrar_Reporte”.

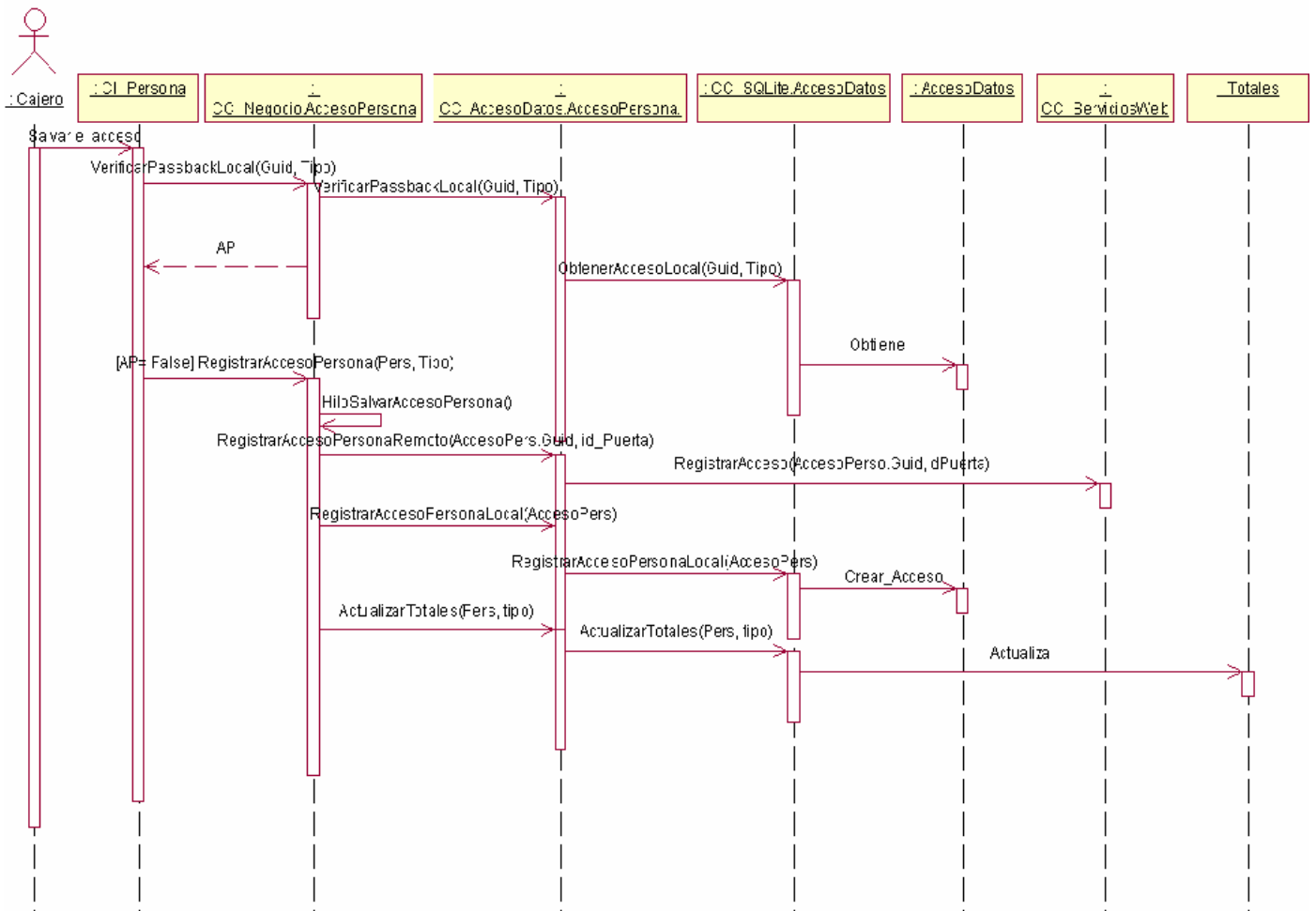


Figura 39. Diagrama de Secuencia del Caso de Uso “Registrar_Accesos”.

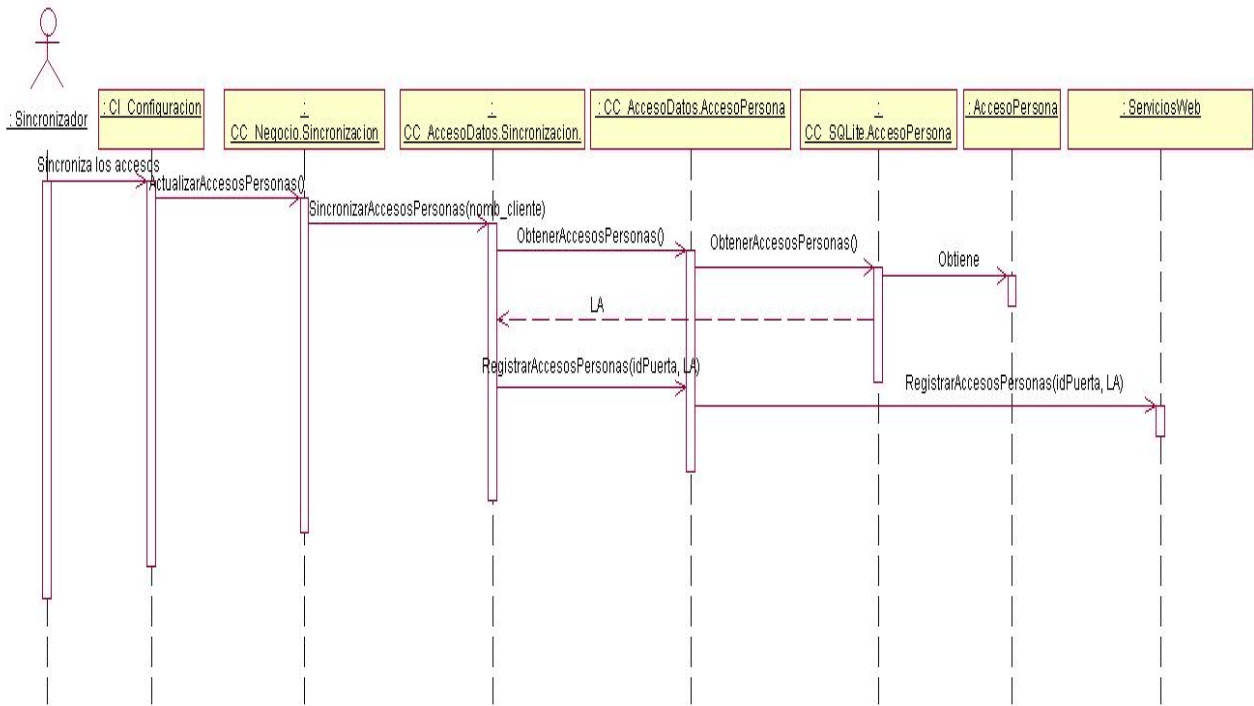


Figura 40. Diagrama de Secuencia del Caso de Uso “Sincronizar_Acceso_Persona”.

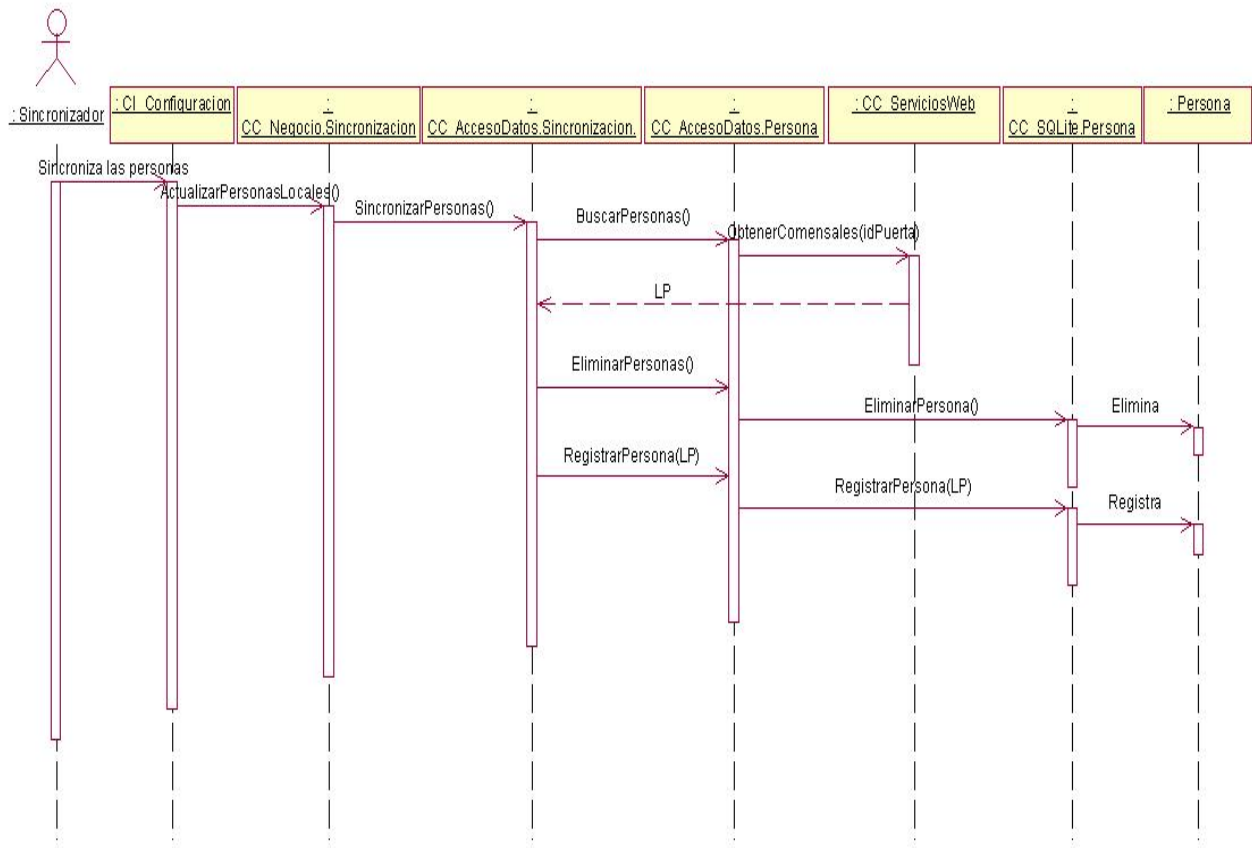


Figura 41. Diagrama de Secuencia del Caso de Uso “Sincronizar_Persona”.

Anexo 4.**Descripción de las tablas de la base de datos.**

A continuación se describen cada una de las tablas mostradas en el diagrama entidad relación de la base de datos.

Nombre: Persona		
Descripción: Almacena los datos de las personas autorizadas a pasar por la puerta.		
Atributo	Tipo	Descripción
Guid	String	Es un identificador para cada persona
NombreCompleto	String	Contiene el nombre completo de la persona
Solapin	String	Campo que contiene el número de solapín de cada persona, es único
CodigoBarras	String	Campo que contiene el código de barras asignado a cada persona, es único
Tipo	String	Guarda el tipo de persona que es (estudiante, trabajador, etc.)
IdTipo	String	Es un identificador del tipo de persona
DireccionFoto	String	Contiene la URL donde se encuentran las fotos de las personas
Fecha	DateTime	Almacena la fecha de sincronización de personas
NombreCargo	String	Guarda el cargo que tienen las personas
Area	String	Guarda el área a la que están vinculadas las personas

Tabla 27. "Persona"

Nombre: AccesosPersona		
Descripción: Guarda los accesos de las personas que fueron registrados.		
Atributo	Tipo	Descripción
IdAcceso	Integer	Es un identificador para los accesos
Guid	String	Es un identificador para cada persona.
IdPuerta	Integer	Guarda el identificador de la puerta en la cual se realizó el acceso
FechaHora	Date Time	Guarda fecha y hora del registro del acceso

Tabla 28. "AccesosPersona".

Nombre: Usuario		
Descripción: Guarda el usuario que tiene permisos para acceder a la configuración del sistema.		
Atributo	Tipo	Descripción
Usuario	String	Almacena usuario

Pass	String	Guarda contraseña
------	--------	-------------------

Tabla 29. "Usuario".

Nombre: Configuracion		
Descripción: Almacena todos los datos de la última vez que se realizó algún cambio en la configuración, estos datos no serán reemplazados hasta la próxima actualización.		
Atributo	Tipo	Descripción
llave	Varchar(50)	Guarda los identificadores correspondientes a los datos que se cambian en la configuración, son valores únicos
valor	Varchar(50)	Contiene aquellos valores que se van actualizando, son valores temporales, pues se irán cambiando a medida que se actualicen los mismos en la configuración

Tabla 30. "Configuración".

Nombre: Puerta		
Descripción:		
Atributo	Tipo	Descripción
IdPuerta	String	Es un identificador para las puertas
Descripción	Varchar(50)	Contiene información sobre la puerta
EstaUso	Bool	Guarda si está activa la puerta o si no lo está

Tabla 31. "Puerta".

Nombre: Totales		
Descripción:		
Atributo	Tipo	Descripción
IdTipoPersona	Integer	Es un identificador del tipo de persona.
TipoPersona	Varchar(50)	Guarda el tipo de persona que es (estudiante, trabajador, etc.).
CantidadAsignados	Integer	Almacena la cantidad de personas por tipos que pueden pasar por la puerta.
CantidadAccedidos	Integer	Almacena la cantidad de personas accedidas por tipo que pueden pasar por la puerta.
CantidadDenegados	Integer	Almacena la cantidad de personas denegadas por tipo.

Tabla 32. "Totales".

Glosario de Términos

CCD (Charge Coupled Device): Dispositivo de carga emparejada, instrumento cuyos semiconductores están conectados de tal manera que la salida (output) de uno sirve como entrada (input) del próximo.

CLR (Common Language Runtime): Es un entorno de ejecución en el que aplicaciones escritas en diferentes lenguajes pueden trabajar juntas, esto se consigue a través de lo que se denomina código gestionado, que no es más que el código de las aplicaciones desarrolladas para la plataforma .NET, es decir, el CLR se encarga de gestionar la ejecución de estas aplicaciones, a las que además ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad.

CLS (Common Language Specification o Especificación del Lenguaje Común): Es un conjunto de reglas que han de seguir las definiciones de tipos usando un determinado lenguaje gestionado si se desea que sean accesibles desde cualquier otro lenguaje gestionado.

HTTP (Hyper Text Transfer Protocol): Protocolo de Transferencia de Hipertexto es el protocolo que se utiliza para incursionar en los sitios WWW en Internet, usado en cada transacción. El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página Web, y la respuestas, remitiendo la información que se verá en pantalla. También se envía información adicional en ambos sentidos, como formularios con mensajes y otros similares.

MathML (Mathematical Markup Language): Es un lenguaje de marcado basado en XML, cuyo objetivo es expresar notación matemática de forma que distintas máquinas puedan entenderla, para su uso en combinación con XHTML en páginas web, y para intercambio de información entre programas de tipo matemático en general.

El soporte de MathML es grande en programas matemáticos (como Maple o Mathcad), pero aún escaso en navegadores (sólo vía plugins en Internet Explorer, parcial en Mozilla).

GLOSARIO DE TÉRMINOS

Password (Una contraseña (*password* en inglés) o clave): Es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso. La contraseña normalmente debe mantenerse en secreto ante aquellos a quien no se les permite el acceso. Aquellos que desean acceder a la información se les solicita una clave; si conocen o no conocen la contraseña, se concede o se niega el acceso a la información según sea el caso.

RAD (Rapid Application Development): Desarrollo rápido de aplicaciones, conjunto de instrumentos para el desarrollo de programas que facilita al programador desarrollar un programa como un sistema de componentes que puede actualizar en forma rápida y adaptarlo fácilmente a nuevas tecnologías (Informática).

SOAP (Simple Object Access Protocol): Es un protocolo estándar creado por Microsoft, IBM y otros. Define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los Servicios Web, ya que es el mecanismo estándar de empaquetar los mensajes para los Servicios Web. Consiste de tres partes: la primera define cuál es el mensaje y cómo procesarlo, la segunda es un sistema de reglas de codificación para expresar tipos de datos definidos y una tercera parte para representar respuestas de llamadas por parte de procedimientos remotos.

Stylus (Aguja fonográfica): Punta de metal o zafiro que recorre los surcos de los discos fonográficos para reproducir las vibraciones inscritas en ellos.

SVG (Scalable Vector Graphics): Es un lenguaje para describir gráficos vectoriales bidimensionales, tanto estáticos como animados (estos últimos con ayuda de SMIL), en XML. SVG se convirtió en una Recomendación del W3C en Septiembre de 2001, por lo que ya ha sido incluido de forma nativa en el navegador Web del W3C **Amaya**. Otros navegadores Web, necesitan un conector o plug-in, para lo que se puede utilizar el Visualizador SVG de Adobe.

GLOSARIO DE TÉRMINOS

USB (Universal Serial Bus): Bus de Serie Universal es un canal para el traspaso de datos universal, interfase que provee un estándar de bus serie para conectar dispositivos a un ordenador personal (generalmente a un PC).

WSDL (Web Services Description Language): Es un formato XML que se utiliza para describir Servicios Web. WSDL describe la interfaz pública a los Servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

XHTML (eXtensible Hyper Text Markup Language o lenguaje extensible de marcado de hipertexto): Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una Web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo (como las hojas de estilo en cascada) y JavaScript su aspecto y diseño en distintos medios (ordenadores, PDAs, teléfonos móviles, impresoras).

XML (Extensible Markup Language o Lenguaje extensible de etiquetas): Es un metalenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.