

Universidad de Ciencias Informáticas UCI

Facultad I

**Trabajo de Diploma para optar por el título de  
INGENIERO EN CIENCIAS INFORMÁTICAS**

**Título: Módulo de administración para la Plataforma  
de Identificación de la UCI**

---

**Autores**

Dailen Barrera Millet

Leonardo Díaz Novo

**Tutor**

Ing. Reynier Lester Claro Escalona

**Ciudad de La Habana, Cuba**

**26 de Junio del año 2013**

**“Año 54 de la Revolución”**



*“Mejor es adquirir sabiduría que oro preciado; y adquirir inteligencia  
vale más que la plata.”*

*Proverbio del rey Salomón*

## DECLARACIÓN DE AUTORÍA

Se declara que los únicos autores de este trabajo son Dailen Barrera Millet y Leonardo Díaz Novo y se autoriza a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste se firma a los \_\_\_\_ días del mes de \_\_\_\_ del año \_\_\_\_ el presente documento.

\_\_\_\_\_

Dailen Barrera Millet

\_\_\_\_\_

Leonardo Díaz Novo

\_\_\_\_\_

Ing. Reynier Lester Claro Escalona

## SÍNTESIS DEL CONTACTO

**Nombre y Apellidos:** Reynier Lester Claro Escalona.

Ingeniero en Ciencias Informáticas, UCI 2008.

**Categoría Docente:** Profesor Instructor del Departamento de Identificación del Centro de Identificación y Seguridad Digital (CISED), Facultad 1.

Arquitecto del Proyecto Plataforma Modular de Identificación y Control de Acceso, del Departamento de Identificación del CISED.

**Institución a la que pertenece:** Universidad de las Ciencias Informáticas (UCI).

**Dirección:** Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

**Correo electrónico:** rlclaro@uci.cu.

## RESUMEN

En la Universidad de Ciencias Informáticas (UCI), ante la necesidad de identificar al personal que allí reside mediante un documento que los acredite como personal autorizado, el departamento de Identificación del Centro de Identificación y Seguridad Digital (CISED) pretende desarrollar una Plataforma Modular de Identificación y Control de Acceso (PMICA). Dicha plataforma se compone de dos subsistemas: Identificación y Control de Acceso, cada uno compuesto por varios módulos independientes. Para el buen funcionamiento de estos módulos es necesaria su integración y realizar las configuraciones globales necesarias de los procesos que maneja cada subsistema de la plataforma.

Para ello se propone la creación de un módulo de administración que gestione los diferentes elementos de configuración y seguridad de ambos subsistemas, para el correcto funcionamiento de la PMICA.

El presente trabajo describe el desarrollo del módulo de administración para el Subsistema de Identificación, convirtiéndose este último en la Plataforma de Identificación de la UCI.

Para el desarrollo del mismo se realiza un estudio de soluciones existentes, se hace una propuesta de solución y se procede a realizar el análisis, diseño e implementación de dicha solución. Para ello fue necesario el uso de un conjunto de herramientas y tecnologías modernas que garanticen la eficiencia y calidad del trabajo.

Con la implantación de este módulo en el sistema se podrán realizar las funcionalidades de configuración y seguridad de los distintos módulos de la Plataforma de Identificación de la UCI, posibilitando que el sistema sea amigable, flexible, fácil de controlar y usar.

**Palabras claves:** Identificación, Plataforma, Módulo, Subsistema, Configuración, Seguridad.

# ÍNDICE

DECLARACIÓN DE AUTORÍA .....	III
SÍNTESIS DEL CONTACTO .....	IV
RESUMEN .....	V
ÍNDICE .....	VI
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS.....	X
INTRODUCCIÓN.....	12
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	17
1.1 Introducción del Capítulo.....	17
1.2 Conceptos Fundamentales .....	17
1.3 Estudio de Soluciones Existentes .....	19
1.3.1 Necesidad de la creación de un módulo de administración para la Plataforma de Identificación de la UCI.....	23
1.4 Ambiente de Desarrollo .....	24
1.5 Conclusiones del Capítulo.....	35
CAPÍTULO II: PROPUESTA DE SOLUCIÓN.....	36
2.1 Introducción del Capítulo.....	36
2.2 Modelo de Dominio.....	36
2.2.1 Glosario de Conceptos del Modelo de Dominio .....	37
2.2.2 Descripción del Modelo de dominio .....	38
2.3 Concepción de los módulos del sistema.....	39
2.4 Especificación de los Requerimientos del <i>Software</i> .....	40
2.4.1 Requisitos Funcionales del Sistema .....	40
1.4.2 Descripción de los Requisitos Funcionales .....	42
2.4.2 Requisitos No Funcionales del Sistema .....	47
2.5 Arquitectura de la Solución .....	49
2.6 Diagrama de Clases del Diseño.....	51
2.7 Diagrama de Secuencia .....	53
2.8 Diseño de la Base de Datos .....	54

2.8.1 Modelo de Datos .....	54
2.8.2 Descripción del Modelo de Datos .....	54
2.9 Patrones de Diseño .....	56
2.10 Elementos de Seguridad en el Sistema.....	57
2.11 Conclusiones del Capítulo.....	58
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA .....	59
3.1 Introducción .....	59
3.2 Implementación .....	59
3.2.1 Diagrama de Despliegue.....	59
3.2.2 Diagrama de Componentes .....	60
3.2.3 Estándares de Codificación.....	62
3.2.4 Tratamiento de Errores .....	64
3.3 Pruebas de <i>Software</i> .....	65
3.3.1 Pruebas de Caja Blanca (estructural).....	66
3.3.2 Pruebas de Caja Negra (funcional).....	67
3.3.3 Resultado General de las Pruebas .....	72
3.4 Conclusiones Parciales.....	72
CONCLUSIONES .....	73
RECOMENDACIONES .....	74
BIBLIOGRAFÍA REFERENCIADA.....	75
BIBLIOGRAFÍA CONSULTADA.....	77
GLOSARIO DE TÉRMINOS.....	78
ANEXOS .....	80
ANEXO 1. Interfaces de los sistemas estudiados.....	80
ANEXO 2. Descripción de las funcionalidades.....	84
ANEXO 3. Diagramas de clases del diseño .....	110
ANEXO 4. Descripción de las entidades del Modelo de Datos.....	114
ANEXO 5. Diagramas de secuencia.....	118
ANEXO 6. Casos de prueba de caja negra .....	122
ANEXO 7. Resultado de las pruebas de caja blanca .....	129

## ÍNDICE DE FIGURAS

<i>Figura 1 Modelo de dominio del módulo de administración</i> .....	37
<i>Figura 2 Concepción de los módulos</i> .....	39
<i>Figura 3 Vista lógica de la arquitectura de software</i> .....	50
<i>Figura 4 Diagrama de clases "Gestionar Rol"</i> .....	52
<i>Figura 5 Diagrama de secuencia "Autenticar Usuario"</i> .....	53
<i>Figura 6 Modelo de datos</i> .....	54
<i>Figura 7 Ejemplo del uso del patrón creador</i> .....	56
<i>Figura 8 Ejemplo del uso del patrón singleton</i> .....	57
<i>Figura 9 Diagrama de despliegue</i> .....	59
<i>Figura 10 Diagrama de Componentes</i> .....	61
<i>Figura 11 Respuesta del sistema al entrar letras en el campo de ci del ciudadano</i> .....	65
<i>Figura 12 Respuesta de sistema al asignar un puesto de trabajo a un usuario que no existe</i> .....	65
<i>Figura 13 Resultado de la prueba unitaria realizada a la funcionalidad CrearOficina()</i> .....	67
<i>Figura 14 Iteraciones de la prueba de caja negra</i> .....	72
<i>Figura 15 Interfaz principal del sistema SAIME</i> .....	80
<i>Figura 16 Interfaz "Crear Dirección" del sistema SUIN</i> .....	81
<i>Figura 17 Interfaz "Autenticar" del Sistema de Identificación de la UCI</i> .....	81
<i>Figura 18 Interfaz "Gestionar Roles" del Sistema de Identificación de la UCI</i> .....	82
<i>Figura 19 Interfaz "Gestionar Usuarios" del Sistema de Identificación de la UCI</i> .....	83
<i>Figura 20 Interfaz "Gestión de Estado de Credencial" del Sistema de Identificación de la UCI</i> .....	83
<i>Figura 21 Diagrama de clases del diseño "Gestionar Usuario"</i> .....	110
<i>Figura 22 Diagrama de clases del diseño "Gestionar Rol"</i> .....	111
<i>Figura 23 Diagrama de clases del diseño "Gestionar Oficinas"</i> .....	112
<i>Figura 24 Diagrama de clases del diseño "Gestionar Puestos de Trabajo"</i> .....	113
<i>Figura 25 Diagrama de clases del diseño Registro de Operaciones</i> .....	114
<i>Figura 26 Diagrama de secuencia "Gestionar Ciudadano". Escenario "Buscar Ciudadano"</i> .....	118
<i>Figura 27 Diagrama de secuencia "Gestionar Ciudadano". Escenario "Crear Ciudadano"</i> .....	119



<i>Figura 28 Diagrama de secuencia "Gestionar Usuario". Escenario "Editar Usuario" .....</i>	<i>120</i>
<i>Figura 29 Diagrama de secuencia "Asignar Rol a Usuario" .....</i>	<i>120</i>
<i>Figura 30 Diagrama de secuencia "Gestionar Oficina". Escenario "Crear Oficina" .....</i>	<i>121</i>
<i>Figura 31 Diagrama de secuencia "Gestionar Oficina". Escenario "Actualizar Oficina" ...</i>	<i>121</i>
<i>Figura 32 Diagrama de secuencia "Configurar Aplicación" .....</i>	<i>122</i>
<i>Figura 33 Resultado de la prueba unitaria realizada a la funcionalidad CrearUsuarioRol() .....</i>	<i>129</i>
<i>Figura 34 Resultado de la prueba unitaria realizada a la funcionalidad CrearRol() .....</i>	<i>130</i>

## ÍNDICE DE TABLAS

<i>Tabla 1 Tabla comparativa de los sistemas estudiados.....</i>	<i>23</i>
<i>Tabla 2 Descripción de la funcionalidad "Gestionar Rol" .....</i>	<i>47</i>
<i>Tabla 3 Descripción de las clases del diagrama de clases "Gestionar Rol" .....</i>	<i>53</i>
<i>Tabla 4 Composición de tablas del modelo de datos .....</i>	<i>55</i>
<i>Tabla 5 Uso de las convenciones Camel y Pascal .....</i>	<i>62</i>
<i>Tabla 6 Escenario "Añadir Rol" del caso de prueba "Gestionar Rol" .....</i>	<i>70</i>
<i>Tabla 7 Escenario "Modificar Rol" del caso de prueba "Gestionar Rol" .....</i>	<i>70</i>
<i>Tabla 8 Escenario "Mostrar Información" del caso de prueba "Gestionar Rol" .....</i>	<i>71</i>
<i>Tabla 9 Descripción de las variables del caso de prueba "Gestionar Rol" .....</i>	<i>71</i>
<i>Tabla 10 Descripción de la funcionalidad "Autenticar Usuario" .....</i>	<i>86</i>
<i>Tabla 11 Descripción de la funcionalidad "Gestionar Ciudadano" .....</i>	<i>90</i>
<i>Tabla 12 Descripción de la funcionalidad "Gestionar Usuario" .....</i>	<i>93</i>
<i>Tabla 13 Descripción de la funcionalidad "Asignar Permiso a Rol" .....</i>	<i>95</i>
<i>Tabla 14 Descripción de la funcionalidad "Eliminar Permiso" .....</i>	<i>96</i>
<i>Tabla 15 Descripción de la funcionalidad "Asignar Rol a Usuario" .....</i>	<i>99</i>
<i>Tabla 16 Descripción de la funcionalidad "Gestionar Oficina" .....</i>	<i>102</i>
<i>Tabla 17 Descripción de la funcionalidad "Gestionar Puestos de Trabajo" .....</i>	<i>105</i>
<i>Tabla 18 Descripción de la funcionalidad "Configurar Aplicación" .....</i>	<i>107</i>
<i>Tabla 19 Descripción de la funcionalidad "Registro de Operaciones" .....</i>	<i>109</i>
<i>Tabla 20 Descripción de la funcionalidad "Editar Perfil de Usuario del Sistema" .....</i>	<i>110</i>
<i>Tabla 21 Entidad correspondiente al ciudadano .....</i>	<i>115</i>
<i>Tabla 22 Entidad correspondiente al tipo de ciudadano .....</i>	<i>115</i>
<i>Tabla 23 Entidad correspondiente al usuario .....</i>	<i>115</i>
<i>Tabla 24 Entidad correspondiente al rol .....</i>	<i>116</i>
<i>Tabla 25 Entidad correspondiente a la relación usuario-rol .....</i>	<i>116</i>
<i>Tabla 26 Entidad correspondiente a los permisos .....</i>	<i>116</i>
<i>Tabla 27 Entidad correspondiente a las oficinas .....</i>	<i>117</i>
<i>Tabla 28 Entidad correspondiente a los tipos de puesto .....</i>	<i>117</i>
<i>Tabla 29 Entidad correspondiente a los puestos de trabajo .....</i>	<i>117</i>
<i>Tabla 30 Entidad correspondiente a los reportes de operaciones .....</i>	<i>118</i>
<i>Tabla 31 Entidad correspondiente al tipo de acción .....</i>	<i>118</i>

<i>Tabla 32 Escenario "Buscar Ciudadano" del caso de prueba "Gestionar Ciudadano" .....</i>	<i>123</i>
<i>Tabla 33 Escenario "Crear Ciudadano" del caso de prueba "Gestionar Ciudadano" .....</i>	<i>124</i>
<i>Tabla 34 Escenario "Asignar Rol" del caso de prueba "Asignar Rol a Usuario" .....</i>	<i>125</i>
<i>Tabla 35 Escenario "Eliminar Asignación" del caso de prueba "Asignar Rol a Usuario" ..</i>	<i>126</i>
<i>Tabla 36 Escenario "Añadir Oficinal" del caso de prueba "Gestionar Oficina" .....</i>	<i>128</i>
<i>Tabla 37 Escenario "Gestionar Oficina" del caso de prueba "Gestionar Oficina" .....</i>	<i>128</i>

## INTRODUCCIÓN

Desde el origen del hombre, la necesidad de “identificar” es innegable en cualquier ámbito, tanto en personas como en objetos y animales. Sus primeros antecedentes se remontan a las tribus y clanes de la antigüedad, donde se reconocían mutuamente a través de la fisonomía. Con la llegada de los asentamientos y la distribución de la población surge la necesidad de los nombres y luego de los apellidos, para diferenciar a las personas con un mismo nombre.

En nuestros días, identificar una persona, enseñar un documento de identidad, son eventos del mundo contemporáneo, el cual adapta sus mecanismos de identificación a la realidad moderna que se impone con la llegada de la sociedad de la información y la generalización del uso de Internet.

Los sistemas informáticos actuales son cada vez más flexibles y adaptables en dependencia de las necesidades de los clientes. Por lo general están compuestos por varios módulos independientes que realizan determinadas funciones y todos ellos unidos por una estructura organizativa llamada módulo de administración, que permitirá que el sistema funcione de forma ordenada, segura y consistente.

En la Universidad de Ciencias Informáticas (UCI) existe igualmente la necesidad de identificar al personal que allí reside (profesores, trabajadores y estudiantes) con el fin de llevar un control de los mismos. Para ello es necesario que cada individuo porte una credencial (documento de identificación) que lo acredite como personal autorizado en la universidad.

La UCI cuenta con el Centro de Identificación y Seguridad Digital (CISED) para el desarrollo de soluciones integrales, productos y servicios en el campo de la identificación y la seguridad digital. En el departamento de Identificación perteneciente a dicho centro, se ejecutan proyectos relacionados con la implantación de soluciones integrales de identificación y control de fronteras. El departamento pone en marcha el desarrollo de la Plataforma Modular de Identificación y Control de Acceso (PMICA) el cual tiene como objetivo proveer un mecanismo de identificación para las personas en la UCI y adecuarse a las diferentes necesidades y condiciones de los clientes de cualquier empresa. LA PMICA estará compuesta por dos subsistemas: Subsistema de

Identificación y Subsistema de Control de Acceso.

Cada uno de ellos estará compuesto por diferentes módulos con el fin de agrupar las funcionalidades más importantes, logrando de esta forma una mejor estructura y organización.

El Subsistema de Identificación es el encargado de llevar todo el proceso de identificación del personal que reside en la universidad, desde la solicitud de un documento de identificación, captura de datos, captura de imágenes, revisión de datos e imágenes, impresión de documentos de identificación, control de la calidad, y termina con la entrega de dicho documento.

El estudio realizado al subsistema arrojó que el mismo requiere un control minucioso de las configuraciones globales, (dígase seguridad, accesibilidad y la gestión de elementos como la fuente de donde se obtendrán los datos, entre otros) para un mejor desarrollo de estos procesos. Se encontró además que en el subsistema existen funcionalidades comunes para algunos módulos que de desarrollarse las mismas de manera repetida, se incumpliría con los principios de reutilización del código. El subsistema necesita además gestionar los niveles de accesos de cada uno de los usuarios responsables de alguna tarea específica, sin lo cual no pudiera considerarse una herramienta segura.

Por estas razones existe la necesidad de contar con un módulo encargado de gestionar los diferentes elementos de configuración y seguridad para el funcionamiento del subsistema, convirtiéndose este último en la Plataforma de Identificación de la UCI.

A partir de la situación problemática antes expresada se define como **problema de investigación** ¿Cómo gestionar los elementos de configuración y seguridad de la Plataforma de Identificación de la UCI?

Quedando definido como **objeto de estudio** Los procesos de configuración y administración en sistemas de gestión de información.

Se plantea como **objetivo general**: Desarrollar el módulo de administración para la Plataforma de Identificación de la UCI. Quedando desglosado en los siguientes **objetivos específicos**:

1. Desarrollar el marco teórico de la investigación.
2. Realizar análisis y diseño para el módulo de administración.
3. Implementar el módulo de administración utilizando tecnología *web*.

4. Probar la solución implementada del módulo de administración.

El **campo de acción** estará enfocado en el proceso de administración de la Plataforma de Identificación de la UCI. Para dar cumplimiento al objetivo planteado se definen las siguientes **tareas de investigación**:

1. Análisis de los principales conceptos asociados a la identificación y administración para obtener una base teórica necesaria para el desarrollo de la solución.
2. Caracterización de las soluciones informáticas de administración en sistemas de identificación.
3. Definición de las herramientas informáticas y metodologías a usar para el desarrollo módulo de administración.
4. Diseño del modelo de dominio del módulo de administración.
5. Definición de los requisitos funcionales y no funcionales del módulo de administración.
6. Descripción de los requisitos funcionales y no funcionales del módulo de administración.
7. Diseño de prototipos de interfaz de usuario.
8. Descripción de los requisitos del módulo de administración.
9. Diseño de las clases del módulo de administración.
10. Implementación de los componentes necesarios para el módulo de administración.
11. Validación de la propuesta de solución aplicando al sistema pruebas de caja negra y de caja blanca.

La presente investigación propone la siguiente **idea a defender**: Con el desarrollo de un módulo de administración se podrán gestionar los elementos de configuración y seguridad de la Plataforma de Identificación de la UCI.

Para realizar de forma efectiva el cumplimiento de las tareas planteadas, se utilizarán los siguientes **métodos científicos**:

## **Métodos teóricos:**

1. Análisis histórico-lógico: Permite analizar la trayectoria completa de la gestión de la información que se lleva a cabo en el proceso de identificación en la UCI, así como el estudio histórico del mismo, que permite ver deficiencias y proponer soluciones de acuerdo a las necesidades.
2. Analítico sintético: Mediante este método se analiza toda la información recopilada en los medios bibliográficos que pueda servir para desarrollar mejor el diseño del sistema, y aplicar estos conocimientos para tener una mejor preparación en el tema.

## **Métodos empíricos:**

1. Observación: Permite visualizar lo que ocurre realmente en el proceso de emisión de documentos de identificación en la UCI, con el objetivo de clasificar las dificultades existentes y las funcionalidades que debe tener el sistema a implementar.
2. Entrevista: Este método se utiliza para obtener toda la información referente a los problemas de seguridad y administración de la información en el proceso de emisión de documentos de identificación.
3. Revisión documental: Se pone de manifiesto al revisar la bibliografía existente y consultar la información en Internet para llevar a cabo las tareas de investigación.

## **Justificación de la Investigación:**

Con la presente investigación se contará con un módulo de administración que permitirá realizar las funcionalidades de configuración y seguridad de los distintos módulos de la Plataforma de Identificación de la UCI, posibilitando que el sistema sea amigable, flexible, fácil de controlar y usar.

El contenido del presente trabajo está estructurado en tres capítulos distribuidos de la siguiente manera:

**CAPITULO I:** "Fundamentación teórica": Se describen los conceptos fundamentales que serán tratados a lo largo de la investigación, así como el estudio de sistemas de identificación que implementan su administración. Se hace un análisis de las diversas metodologías y tecnologías de desarrollo de *software* para su uso en el ciclo de desarrollo de la aplicación.

**CAPITULO II:** "Propuesta de Solución": Se exponen las principales características del sistema y se realiza una propuesta general. Se define una lista de funcionalidades que son descritas detalladamente para guiar la implementación de la solución. Se realiza la planeación de las iteraciones que se llevarán a cabo para diseñar y construir las funcionalidades. Se propone el diseño de las funcionalidades.

**CAPITULO III:** "Implementación y Prueba": Dedicado a la implementación de las funcionalidades definidas en el capítulo anterior. Se realizan las pruebas al módulo de administración, validando así la solución propuesta a través de los resultados obtenidos.



## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción del Capítulo

En el desarrollo de este capítulo se abordan algunos conceptos y términos relacionados con el dominio del problema. Se presenta el resultado del estudio realizado sobre algunos sistemas de identificación que implementan un módulo de administración. Se hace la selección de la metodología de desarrollo a utilizar durante todo el ciclo de desarrollo, además de las herramientas para el modelado e implementación de la aplicación, fundamentando las ventajas del uso de cada una de ellas.

### 1.2 Conceptos Fundamentales

En el desarrollo del trabajo se utilizan algunos términos técnicos, de los cuales a continuación se exponen sus definiciones para un mejor entendimiento y comprensión de los mismos.

#### **Identificación**

El término identificación se usa para designar al acto de identificar, reconocer o establecer los datos e información principal sobre una persona. La identificación también abarca los documentos que sirven para identificar a cada persona de forma oficial y legal. Papeles tales como los documentos de identidad, pasaportes, licencias de conducir o cédulas, carnets de diferentes tipos y hasta tarjetas de consumo, sirven como elementos para dar identidad a cada individuo. (1)

#### **Sistemas Informáticos de Identificación**

Un sistema de identificación es cualquier aplicación o *software* que haga uso de uno o varios mecanismos que sirvan para identificar a las personas. Por lo general contienen toda la información necesaria para verificar la identidad del individuo y permitirle el acceso a la información según los privilegios correspondientes al rol que el mismo tenga asignado.

# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

## **Módulo**

En programación un módulo es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso).

Los módulos suelen estar (no necesariamente) organizados jerárquicamente en niveles, de forma que hay un módulo principal que realiza las llamadas oportunas a los módulos de nivel inferior.

(3)

### ***Características de un módulo***

Cada uno de los módulos de un programa idealmente debería cumplir las siguientes características:

- ✓ **Tamaño relativamente pequeño:** Esto facilita aislar el impacto que pueda tener la realización de un cambio en el programa, para corregir un error, o por rediseño del algoritmo correspondiente.
- ✓ **Independencia modular:** Cuanto más independientes son los módulos entre sí más fácil y flexiblemente se trabajará con ellos, esto implica que para desarrollar un módulo no es necesario conocer detalles internos de otros módulos.

Como consecuencia de la independencia modular un módulo cumplirá:

- Características de caja negra, o sea abstracción.
- Aislamiento de los detalles mediante encapsulamiento. (3)

## **Administración**

Se denomina administración al conjunto de las funciones o procesos básicos (planificar, organizar, dirigir, coordinar y controlar) que, realizados convenientemente, repercuten de forma positiva en la eficacia y eficiencia de una actividad realizada. (4)

### **Módulo de Administración**

Un módulo de administración es una estructura organizativa que permite que el sistema funcione de forma ordenada, segura y consistente. Su uso garantiza eficiencia en el resto de los módulos y

# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

sus procesos, permitiendo la centralización de algunos procedimientos y configuraciones globales que garantizarían el buen funcionamiento del mismo. Realizan varias operaciones fundamentales:

- ✓ Controlar las operaciones como parte de la seguridad.
- ✓ Definir las estructuras lógicas para tener una buena organización en el sistema.
- ✓ Gestionar las configuraciones globales o específicas de los negocios que integra.

## ***¿Qué tipo de funcionalidades debe realizar un módulo de administración?***

El módulo de administración de un sistema se encarga de realizar funcionalidades de carácter administrativo con el objetivo de perfeccionar todo el funcionamiento del mismo, como son:

Gestión o configuración de los usuarios, la oficina donde operan sus funcionarios, gestión de grupos de usuarios, definición de servidores, definición de bases de datos a acceder, definición de dominios, gestión de sesiones activas y estadísticas para balanceo de carga y mejoramiento del tiempo de respuesta, buscador de objetos, herramientas de colaboración entre desarrolladores, control de acceso restringido para cada proyecto, y otras funcionalidades que son comunes para cualquier sistema. (5)

## **1.3 Estudio de Soluciones Existentes**

En la búsqueda de soluciones para la situación problemática planteada, se realizó un estudio de algunos sistemas de identificación existentes que implementan un módulo de administración. Para esto se tuvieron en cuenta los siguientes aspectos:

- ✓ Disponibilidad de documentación.
- ✓ Acceso al código fuente.
- ✓ Tecnologías de desarrollo.
- ✓ Similitud con el problema a resolver.

# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

Entre los sistemas estudiados se encontraron los siguientes:

## **SAIME (Servicio Administrativo de Identificación Migración y Extranjería)**

SAIME<sup>1</sup> es un sistema realizado con el objetivo de integrar todos los procesos de identificación, migración y extranjería para la República Bolivariana de Venezuela, logrando de esta forma el control distribuido y centralizado de las operaciones y trámites que se realizan en la ONIDEX<sup>2</sup>. El sistema implementa un módulo de administración encargado de la gestión y configuración de usuarios, oficinas móviles, sesiones y puestos de trabajo, servidores, entre otros; además permite la generación de actualizaciones y la autenticación de los usuarios en el sistema. Con el manejo del mismo se obtienen las siguientes ventajas:

- ✓ Se logra la centralización de la estructura organizativa del sistema SAIME.
- ✓ Se incrementa el control de los operarios que realizan los distintos procesos dentro del sistema.
- ✓ Se controla de forma más eficiente el estado de las oficinas que trabajan con el sistema.
- ✓ Se integran de forma eficiente todos los procesos administrativos que se llevan a cabo en el sistema.

El módulo de administración del SAIME se compone de varios submódulos de administración distribuidos por las aplicaciones que se encuentran en las distintas oficinas del sistema. A continuación se muestra en qué consiste cada uno de ellos:

**Administración de Oficina:** Es un submódulo que permite el control de una oficina del sistema y contiene funcionalidades comunes para todos los demás submódulos de administración.

**Administración CPID:** Es un submódulo que contiene funcionalidades comunes de Administración de Oficina y está incluido en la aplicación del Centro de Personalización e Impresión de Documentos.

**Administración de Migración:** Es un submódulo que contiene funcionalidades comunes de Administración de Oficina y está incluido en las aplicaciones de control Migratorio.

---

<sup>1</sup> Servicio Administrativo de Identificación Migración y Extranjería

<sup>2</sup> Oficina Nacional de Identificación y Extranjería

## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

**Administración Sede Central:** Es un submódulo que contiene funcionalidades comunes de Administración de Oficina y está incluido en la aplicación de Sede Central.

**Administración Global:** Es el submódulo de administración más importante, porque permite el control y gestión global de los conceptos organizativos principales del sistema, además, contiene funcionalidades comunes de la Administración de Oficina.

**Administración Común:** Es un módulo administrativo común donde se agrupan un conjunto de funcionalidades que cualquiera de los otros módulos administrativos necesitarán usar en algunas ocasiones para trabajar con el sistema. (6)

De manera general el SAIME es un sistema en el que se desarrollan un conjunto de procesos, que se implementan a través de varias aplicaciones y cada una contiene un módulo administrativo que permite su control. (Ver imagen "*Interfaz principal del sistema SAIME*" del ANEXO 1: "Interfaces de los sistemas estudiados")

### **SUIN (Sistema Único de Identificación Nacional de la República de Cuba)**

SUIN<sup>1</sup> es un sistema desarrollado para la informatización de los procesos llevados a cabo en las oficinas del CIRP<sup>2</sup>. El mismo permite la administración y configuración en sistemas de gestión de información.

El SUIN se compone de siete módulos. Estos tienen procesos que contienen mucha información y responsabilidades que son comunes entre ellos. Por esta razón el sistema se vale de un módulo de administración para la gestión y configuración del resto de los módulos, lográndose de esta manera:

- ✓ Configurar y gestionar los elementos que serán utilizados en la realización de los procesos de cada uno de los módulos.
- ✓ Recoger las funcionalidades principales de administración y seguridad de un sistema, así como aquellas que son indispensables para llevar a cabo cada uno de los procesos en dicho sistema.
- ✓ Llevar a cabo la gestión de los elementos fundamentales en la realización de cada uno de los trámites que se realicen, con el fin de garantizar el control de toda la información referente a los mismos.

---

<sup>1</sup> Sistema Único de Identificación Nacional de la República de Cuba

<sup>2</sup> Carné de Identidad y Registro de la Población

## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

El Módulo de Administración del SUIN posee tres roles fundamentales encargados de garantizar la administración según el nivel de acceso a la gestión de la información, ellos son: Administrador Nacional, Administrador Provincial y Administrador de Unidad.

Con la implantación del Módulo de Administración para el SUIN se logra una estandarización en los procesos de gestión de direcciones de las personas (funcionalidad más importante que implementa este módulo), así como la manipulación normalizada de los nomencladores utilizados y la asignación homogénea de las direcciones a todos los ciudadanos cubanos y extranjeros residentes en el país. (7) (Ver imagen "Interfaz Crear Dirección del sistema SUIN" del ANEXO 1: "Interfaces de los sistemas estudiados")

### **Sistema de Identificación de la UCI**

El Sistema de Identificación de la UCI surge en el año 2008 ante la necesidad de mejorar la aplicación que se usaba anteriormente para llevar a cabo el proceso de emisión de documentos de identificación al personal de la UCI. La misma generaba disímiles problemas, entre ellos que se dependía de una imprenta para imprimir los documentos de identificación, solo se imprimía en pvc<sup>1</sup> siendo este muy costoso, se utilizaba una plantilla fija para el diseño, los códigos de barra eran consecutivos provocando que los números crecieran cada vez más a tal punto que se salieran de la plantilla, los datos no eran fiables ya que se manipulaban a través de un documento Word. En ese momento existían más de 600 personas sin credenciales.

El Sistema de Identificación de la UCI da solución a todos estos problemas. Cuenta con un diseñador de credenciales, es independiente de la fuente de datos y reduce el tamaño de los códigos ya que se genera una permutación de 36 millones por sede. Es una aplicación desktop que se ha estado utilizando desde ese entonces hasta la actualidad en las oficinas de emisión de documentos de la universidad.

Las funcionalidades que el sistema brinda, se encuentran agrupadas en los siguientes módulos y opciones: Configuración (Gestión de sede, Gestión de entrada de datos, Gestión de nomencladores, Gestión de recursos de almacenamiento de fotos), Seguridad (Gestión de usuarios, Gestión de roles), Identificación (Impresión de credenciales, Gestión de recursos de

---

<sup>1</sup> Pvc: Policloruro de vinilo. Material plástico en el que se imprimen las credenciales.

## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

impresión, Gestión de estado de credencial, Gestión de credenciales, Verificación de credenciales, Personas sin credencial, Gestión de tipos de personas, Gestión de fotos) y Administración (Gestión de tipo de credencial, Diseño de tipo de credencial). (Ver imágenes del sistema en el ANEXO 1: "Interfaces de los Sistemas Estudiados", figuras 14-17)

### 1.3.1 Necesidad de la creación de un módulo de administración para la Plataforma de Identificación de la UCI

Teniendo en cuenta los criterios antes mencionados se elaboró una tabla comparativa (ver tabla 1) de los sistemas estudiados que se muestra a continuación:

Sistemas	Disponibilidad de documentación (Sí/No)	Acceso al código fuente (Sí/No)	Tecnologías de desarrollo	Desarrollado para una entidad específica (Sí/No)
<b>SAIME</b>	Sí	Sí	-Aplicación <i>desktop</i> en plataforma .Net -Lenguaje C# -Gestor de BD Oracle	Sí
<b>SUIN</b>	Sí	Sí	- Aplicación <i>web</i> en plataforma .Net -Lenguaje C# -Oracle 11g	Sí
<b>Sistema de Identificación de la UCI</b>	No	Sí	-Aplicación <i>desktop</i> en la plataforma .Net	Sí

**Tabla 1** Tabla comparativa de los sistemas estudiados

Luego del análisis de la tabla de comparaciones al concluir el estudio realizado, se identificaron un número de prestaciones y servicios que los sistemas ofrecen, convirtiéndolos en una solución adecuada, pero a la vez no viable, debido a que en algunos de los casos no se cuenta con documentación sobre el proceso de desarrollo de los mismos y aunque se tiene acceso al código



# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

fuentes y las tecnologías utilizadas se corresponden de cierta manera con las que se dispone para el desarrollo del módulo de administración, estos sistemas responden a las especificaciones de cada uno, y no se ajustan a las necesidades y características requeridas por la PMICA de ser un sistema genérico que pueda adaptarse fácilmente a cambios.

Por lo anteriormente expuesto se hace necesario crear un nuevo módulo de administración que permita la estandarización del subsistema, de manera que el mismo pueda adaptarse con facilidad a las necesidades y características de cualquier entidad que lo solicite; logrando una mejora de la gestión de los elementos de configuración y seguridad.

No obstante se identificaron las características más importantes relacionadas con la administración de los sistemas que podrían ser utilizadas como referencia para la implementación del módulo que se desea construir y su integración con los demás módulos.

## 1.4 Ambiente de Desarrollo

### 1.4.1 Metodología de Desarrollo:

Las metodologías de desarrollo de *software* son un marco de trabajo utilizado para estructurar, planificar y controlar el proceso de desarrollo de *software*. Las mismas brindan un conjunto de procedimientos, herramientas y soporte documental que garantizan la eficiencia del producto final.

Hoy en día existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Para dar una idea de qué metodología se puede utilizar y cuál se adapta más al medio, se realizó un estudio de las que se consideran más importantes como *Rational Unified Process* (RUP), *Extreme Programming* (XP), *Microsoft Solution Framework* (MSF) y *Feature Driven Development* (FDD), siendo esta última la seleccionada para dar solución a nuestro problema.

A continuación se exponen las razones para escoger FDD como metodología de desarrollo del sistema.



# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

## ***Desarrollo Basado en Funcionalidades (FDD):***

FDD es una metodología ágil para el desarrollo de sistemas a corto plazo. Se basa en un proceso de iteraciones cortas de aproximadamente dos semanas que producen un *software* funcional que el cliente y la dirección de la empresa, centro o proyecto pueden ver y monitorear. Estas iteraciones se deciden en base a funcionalidades que son pequeñas partes del *software* con significado para el cliente. Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado y con su utilización se obtienen resultados periódicos y tangibles.

El proceso consiste en cinco pasos secuenciales durante los cuales se diseña y se construye el sistema:

- ✓ **Desarrollo de un modelo general:** Cuando comienza el desarrollo, los expertos del dominio están al tanto de la visión, el contexto y los requerimientos del sistema a construir. Se divide el dominio global en áreas que son analizadas detalladamente. Los desarrolladores construyen un diagrama de clases o de objetos por cada área y se construye un modelo global del sistema.
- ✓ **Construcción de una lista de funcionalidades:** Una funcionalidad constituye pequeñas partes del *software* con significado para el cliente. Se elabora una lista de funcionalidades que resume la funcionalidad general del sistema; la lista es elaborada por los desarrolladores y es evaluada por el cliente y se divide en subconjuntos según la afinidad y la dependencia de las funcionalidades. La lista es finalmente revisada por los usuarios y los responsables para su validación y aprobación.
- ✓ **Planeación por funcionalidad:** Se procede a ordenar los conjuntos de funcionalidades conforme a su prioridad y dependencia y se asigna a los programadores jefes.
- ✓ **Diseño por funcionalidad y construcción por funcionalidad:** Se selecciona un conjunto de funcionalidades de la lista, se procede a diseñar y construir la funcionalidad mediante un proceso iterativo. Una iteración puede tomar de unos pocos días a un máximo de dos

# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

semanas. El proceso iterativo incluye inspección de diseño, codificación, pruebas unitarias, integración e inspección de código. (8)

Al usar una metodología de *software* acorde con el proyecto que se va a desarrollar se obtienen las siguientes ventajas:

- ✓ Proporciona una guía para ordenar las actividades de un equipo.
- ✓ Dirige las tareas de cada desarrollador por separado y del equipo como un todo.
- ✓ Especifica los artefactos que deben desarrollarse.
- ✓ Ofrece criterios para el control y la medición de los productos y actividades del proyecto.

## 1.4.2 Lenguaje Unificado de Modelado (UML):

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de *software*.

Permite la modelación de sistemas con tecnología orientada a objetos. Se puede aplicar en el desarrollo de *software* entregando gran variedad de formas para dar soporte a una metodología de desarrollo de *software*, pero no especifica en sí mismo qué metodología o proceso utilizar. (9)

### Ventajas de UML:

- ✓ Permite modelar sistemas utilizando técnicas orientadas a objetos.
- ✓ Permite especificar todas las decisiones de análisis, diseño e implementación, construyendo así modelos precisos, no ambiguos y completos.
- ✓ Puede conectarse con lenguajes de programación (ingeniería directa e inversa).
- ✓ Permite documentar todos los artefactos de un proceso de desarrollo.

## 1.4.3 Herramientas Case:

CASE<sup>1</sup> puede ser definida como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software*. Son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del desarrollo de sistemas de información, completamente o en alguna de sus fases. (10)

---

<sup>1</sup> Ingeniería de Software Asistida por Computadora

# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

Las herramientas CASE pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de *software*, algunas de ellas son:

- ✓ Hacer el trabajo de diseño de *software* más fácil y agradable.
- ✓ Verificar el uso de todos los elementos en el sistema diseñado.
- ✓ Ayudar en la documentación del sistema.
- ✓ Ayudar en la creación de relaciones en las bases de datos.
- ✓ Generar estructuras de código.
- ✓ Reducción del costo de producción de *software*. (11)

Entre los tipos de herramientas CASE se encuentran las Herramientas CASE de modelado de *software*.

## **Herramientas Case de Modelado de Software**

CASE cuenta con una amplia lista de aplicaciones. Dos de las más usadas son Rational Rose Enterprise y Visual Paradigm.

### ***Rational Rose***

Rational Rose es una herramienta de diseño orientada a objetos, que da soporte al modelado visual y permite representar gráficamente el sistema. Se centra en los casos de uso y se enfoca hacia un *software* de mayor calidad. Emplea un lenguaje estándar común que facilita la comunicación. Proporciona mecanismos para realizar la ingeniería inversa, permite generar código en diversos lenguajes a partir de un diseño en UML. (12)

### ***Visual Paradigm 8.0***

Visual Paradigm es una herramienta visual de ingeniería de *software* para el modelado, que tiene un entorno de trabajo que muestra una colección de menús, barra de herramientas y ventanas que permiten realizar diagramas. Permite confeccionar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Posibilita el uso de varios idiomas, soporta el modelado de sistemas *web*, es un producto de calidad, fácil de instalar, actualizar y posee gran compatibilidad entre ediciones. Posee como peculiaridad sobre el resto de las herramientas que cuenta con una potente funcionalidad para la creación de interfaces de

## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

usuario de las aplicaciones. Es una herramienta que es soportada por cualquier sistema operativo. (13)

Ambas herramientas presentan grandes posibilidades para el modelado de negocio pero se descarta Rational Rose por ser una herramienta privativa y por tanto se selecciona Visual Paradigm como herramienta CASE ya que la misma presenta licencias gratuitas y comerciales. Además es una herramienta multiplataforma, que permite crear de manera sencilla todos los diagramas y de la cual existe mucha documentación.

Los beneficios de utilizar Visual Paradigm como herramienta CASE de modelado de *software* se muestran a continuación:

- Navegación intuitiva entre código y el modelo.
- Generador de documentación y reportes UML/PDF/HTML/MS Word.
- Demanda en tiempo real, modelo incremental de viaje redondo y sincronización de código fuente.
- Superior entorno de modelado visual.
- Soporte completo de notaciones UML.
- Diagramas de diseño automático sofisticado.
- Análisis de texto y soporte de tarjeta CRC.

### 1.4.4 Entorno Integrado de Desarrollo (IDE):

Un Entorno Integrado de Desarrollo (IDE) es una aplicación de *software* que proporciona servicios integrales a los programadores de computadoras para el desarrollo de *software*. Normalmente está compuesto por:

1. Un editor de código fuente
2. Un compilador y/o un intérprete
3. Automatización de generación de herramientas
4. Un depurador

Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, entre otros. En algunos lenguajes, pueden funcionar como un

## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación de forma interactiva, sin necesidad de trabajo orientado a archivos de texto. (14)

### **Microsoft Visual Studio.Net 2010**

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones *web* ASP.NET, Servicios *Web* XML, aplicaciones de escritorio y aplicaciones móviles. Este IDE habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes como Visual Basic, Visual C# y Visual C++. Dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones *web* ASP y servicios *web* XML. (15)

Visual Studio proporciona las herramientas necesarias para diseñar, desarrollar, depurar e implementar aplicaciones *web*, servicios *web* XML y aplicaciones cliente tradicionales.” (16)

### **1.4.5 Lenguaje de Programación C#:**

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. C# combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET. (17) Algunas de las características más representativas de C# se recogen a continuación de forma reducida:

- ✓ **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- ✓ **Modernidad:** Incorpora en el propio lenguaje elementos que son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes hay que simular.
- ✓ **Orientación a objetos:** No admite ni funciones ni variables globales, sino que todo el código y datos han de concretarse dentro de definiciones de tipos de datos, esto permite que se reduzcan problemas por conflictos de nombres y facilita la legibilidad del código.

# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

- ✓ **Orientación a componentes:** La sintaxis de C# permite que se puedan definir propiedades, eventos y atributos; características que tienen que ser simuladas en otros lenguajes.
- ✓ **Seguridad de tipos:** Incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente.
- ✓ **Eficiencia:** El código incluye numerosas restricciones para alcanzar su seguridad y no permite el uso de punteros.

## 1.4.6 Control de Versiones

Al estar trabajando varias personas en el mismo proyecto, se hará necesario utilizar y modificar archivos que serán comunes para todo el grupo. Por ello, es muy recomendable utilizar un sistema de control de versiones, evitando que al ser utilizados simultáneamente los archivos, o en el caso de que sean utilizados por varias personas, estos sean modificados de manera errónea o se estropee el trabajo realizado por otra persona al sobrescribir el archivo.

### Subversion

Subversion es un *software* de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS<sup>1</sup>, el cual posee varias deficiencias. Es *software* libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo. (18)

### Ventajas de Subversion

- ✓ Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- ✓ Puede ser servido mediante Apache. Esto permite que clientes, utilicen Subversion en forma transparente.

---

1 CVC: (Concurrent Versioning System) Al igual que SVN sirve para controlar las versiones y la historia de los proyectos software.

## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

- ✓ Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
- ✓ Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar.

### **Cliente TortoiseSVN**

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. (19)

### **1.4.7 Plataforma .NET**

La plataforma de desarrollo .NET es una nueva tecnología con énfasis en transparencia de redes, con independencia de plataforma, permitiendo un rápido desarrollo de aplicaciones, combinando la informática y las comunicaciones y ofreciendo las herramientas que se necesitan para transformar las aplicaciones *web*.

.NET utiliza los servicios web como un medio para poder interoperar a distintas tecnologías. Permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios. Les da a los desarrolladores las herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones. La idea central que se implementa con la plataforma .NET es la del uso de servicios con el objetivo de simplificar el desarrollo de aplicaciones *web*.

Está diseñada para que se puedan desarrollar componentes de *software* utilizando casi cualquier lenguaje de programación, de forma que lo que se escriba en un lenguaje pueda utilizarse desde cualquier otro de la manera más transparente posible. Esto es, en vez de estar limitados a un único lenguaje de programación, permitir cualquier lenguaje de programación, siempre y cuando



## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

se adhiera a unas normas comunes establecidas para la plataforma .NET en su conjunto. En el corazón de .NET se encuentra .NET Framework. (20)

### **.NET Framework 4.0**

.NET Framework es un componente integral de Windows que admite la compilación y la ejecución de aplicaciones y servicios *web*. Este entorno de ejecución de runtime tiene incorporado el Common Language Runtime (CLR), que proporciona una administración de la memoria y otros servicios del sistema y una biblioteca de clases completa, que permite a los programadores aprovechar el código sólido y confiable para todas las áreas importantes del desarrollo de aplicaciones. Esta biblioteca incluye ADO.NET, ASP.NET, formularios Windows Forms, Windows Presentation Foundation (WPF) y Windows Workflow Foundation (WF). Proporciona un entorno de ejecución administrado, un desarrollo e implementación simplificada y la integración con una gran variedad de lenguajes de programación. .NET Framework está diseñado para cumplir los siguientes objetivos:

- ✓ Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.
- ✓ Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de *software* y los conflictos de versiones.
- ✓ Ofrecer un entorno de ejecución de código que promueva la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
- ✓ Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.
- ✓ Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

Por otra parte la biblioteca de clases es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario hasta aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios *Web Forms* y los servicios *web* XML. (21)



## 1.4.8 Asp.NET Web Forms 4.0

ASP.NET es un marco de desarrollo *web* unificado para la creación de páginas y sitios *web* con HTML, CSS, JavaScript y secuencias de comandos del servidor. (22) Esta tecnología forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el CLR, entre ellos Microsoft Visual Basic y C#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del CLR, seguridad de tipos, herencia, entre otros.

## 1.4.8 Tecnologías Mapeo de Objetos Relacionales (ORM)

ORM (en inglés Object Relational Mapping), es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos y el sistema de base de datos relacional utilizado en el desarrollo de la aplicación.

El utilizar el ORM, nos puede proporcionar ciertas ventajas:

- ✓ **Rapidez en el desarrollo:** La mayoría de las herramientas actuales permiten la creación del modelo por medio del esquema de la base de datos, leyendo el esquema, se crea el modelo adecuado.
- ✓ **Abstracción de la base de datos:** Al utilizar un sistema ORM se consigue separar totalmente del sistema de base de datos que se utilice, y así si en un futuro se debe cambiar de motor de bases de datos, teniéndose la seguridad de que este cambio no afectará al sistema, siendo el cambio más sencillo.
- ✓ **Reutilización:** Permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.
- ✓ **Seguridad:** Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser los SQL injections.
- ✓ **Mantenimiento del código:** Facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos, haciendo que el mantenimiento del código sea mucho más sencillo.

## .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

- ✓ **Lenguaje propio para realizar las consultas:** Estos sistemas de mapeo traen su propio lenguaje para hacer las consultas, lo que hace que los usuarios dejen de utilizar las sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta. (23)

### ***NHibernate***

NHibernate es un *framework* de ORM integrado a la plataforma .Net que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación. Permite almacenar objetos en la base de datos y actualizarlos desde esta más adelante. Traduce el lenguaje orientado a objeto a un lenguaje que la base de datos entienda. Es decir, genera las sentencias necesarias para que se puedan insertar, actualizar, eliminar y cargar datos. De todos los *frameworks* ORM para .NET, NHibernate es probablemente el más maduro y rico en características de todos. (24)

### **1.4.9 Sistemas Gestor de Base de Datos(SGBD)**

Los SGBD están constituidos por un paquete de *software* cuya función es la gestión del acceso a la BD, las operaciones fundamentales son: crear, modificar, eliminar y obtener la estructura asociada al esquema lógico de una base de datos. El objetivo principal es proporcionar un entorno práctico y eficiente a la hora de almacenar y recuperar la información de la base de datos, lo que facilita la gestión de grandes volúmenes de datos.

### ***PostgreSQL 9.1***

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacional, distribuido bajo licencia BSD<sup>1</sup> y con su código fuente disponible libremente. Fue pionero en muchos de los conceptos del sistema Objeto-Relacional actual. Soporta gran parte del SQL estándar y muchas funcionalidades como son: consultas complejas, disparadores, vistas, integridad transaccional, así como el control de versiones concurrentes que es una estrategia de almacenamiento que permite trabajar con grandes volúmenes de datos. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios *web* que atienden un gran número de solicitudes. Ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos,

---

<sup>1</sup> BSD: Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution).

# .:CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA:.

conservando todas las características de estabilidad y rendimiento. Es multiplataforma, disponible en la mayoría de los sistemas operativos. (25)

## ***Características de PostgreSQL.***

- **Objeto-relacionales:** En cada tabla define una clase que permite implementar la herencia entre tablas o clases, funciones y operadores polimórficos.
- **Código abierto:** Debido a la licencia libre, puede ser usado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea de datos, comerciales o académicas.
- **Extensibilidad:** Es una de las características más importantes ya que puede ser ampliado, se pueden añadir nuevos tipos de datos, nuevas funciones y operadores, e incluso nuevos lenguajes de procedimiento y de cliente.
- **Control de Concurrencia Multi-Versión (MVCC):** Es la tecnología que usa para evitar bloqueos innecesarios.
- **Integridad referencial:** Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

## **1.5 Conclusiones del Capítulo**

En el presente capítulo se presentaron conceptos y términos necesarios para el desarrollo del trabajo. Como resultado de la investigación realizada, se concluye que ninguno de los sistemas estudiados permite dar solución al problema planteado, por lo que se hace necesario desarrollar un nuevo módulo que dé solución al problema. Quedaron seleccionadas la metodología, herramientas y tecnologías necesarias para la realización del módulo de administración para la Plataforma de Identificación de la UCI.

## CAPÍTULO II: PROPUESTA DE SOLUCIÓN

### 2.1 Introducción del Capítulo

A partir del estudio del negocio en el que se enmarca el problema en cuestión se realiza un modelo de dominio donde se representan las entidades principales y cómo están relacionadas entre sí. Se detallan las características y necesidades del sistema y se especifican mediante los requerimientos funcionales y los no funcionales. Se define la arquitectura del sistema y se presentan los diagramas de clases, diagramas de secuencia y modelo de datos de la solución propuesta.

### 2.2 Modelo de Dominio

El modelo de dominio representa conceptos de la realidad física. "Permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se enmarca el sistema." (26)

A continuación se muestra el modelo de dominio para el módulo de administración de la Plataforma de Identificación de la UCI, (ver figura 1) mediante el cual se describen las clases más importantes dentro del contexto del sistema.

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

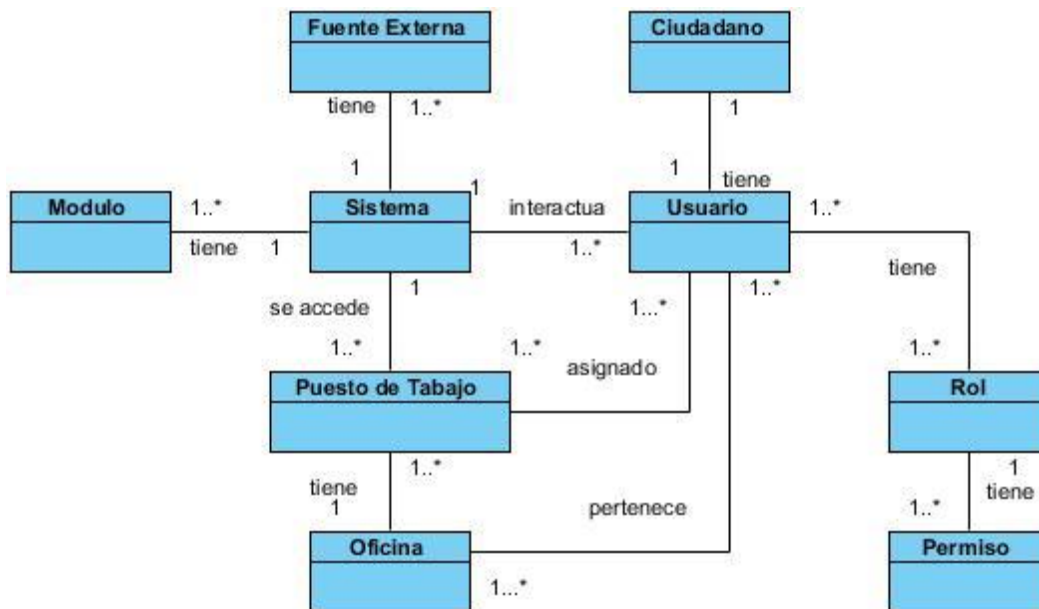


Figura 1 Modelo de dominio del módulo de administración

### 2.2.1 Glosario de Conceptos del Modelo de Dominio

Es necesario tener un vasto conocimiento de cómo debe funcionar el negocio, para poder capturar correctamente los requisitos y así poder construir un sistema con las características que el cliente desea. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Primeramente se identificarán todos los conceptos que se utilizan en el modelo de dominio:

**Sistema:** Es el conjunto de partes interrelacionadas, *hardware*, *software* y de recurso humano que permite almacenar y procesar información, realizado con el objetivo de informatizar el proceso de identificación.

**Módulo:** En programación un módulo es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso). Entre los módulos del sistema se encuentra el módulo de administración el cual gestionará los diferentes elementos de configuración y seguridad del resto de los módulos.

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

**Puesto de trabajo:** Es la estación de trabajo desde donde los usuarios acceden al sistema para interactuar con él.

**Oficina:** Lugar físico donde se lleva a cabo el proceso de emisión de documentos de identificación (solicitud personal, captura de datos, captura de imágenes, revisión de datos e imágenes, impresión de documentos de identificación, control de la calidad y entrega de documentos) En estas áreas los procesos son llevados a cabo por funcionarios con diferentes roles.

**Ciudadano:** Los ciudadanos son las personas que interactúan con el sistema para trabajar con los datos del mismo.

**Usuario:** Es el nombre único que recibe el ciudadano cuando es registrado en el sistema a través del cual podrá autenticarse y dependiendo de sus roles, podrá efectuar determinadas operaciones y tener acceso a los diferentes recursos, en función de los permisos que le sean asignados.

**Rol:** Un rol es una manera de agrupar usuarios acordando a qué partes de la aplicación tienen acceso. Este acceso es asignado o revocado a través de la administración.

**Permiso:** Son asignados a los roles con el objetivo de dar acceso a diferentes recursos en dependencia del rol que posea el usuario.

**Fuente externa:** Lugar de donde se obtendrán los datos a utilizar en el sistema. Se recomienda que sea un LDAP<sup>1</sup>.

### 2.2.2 Descripción del Modelo de dominio

La PMICA es un **sistema** compuesto por varios **módulos** para la realización del proceso de emisión de documentos de identificación. Entre ellos se encuentra el módulo de administración, el cual se encarga de la configuración de elementos globales que utilizan el resto de los módulos. Algunos datos son introducidos al sistema de forma directa, y otros a través de una **fuentes externa** de la cual se alimenta, con el objetivo de obtener los datos necesarios de los ciudadanos. Los **ciudadanos** son agregados al sistema con el mismo **usuario** que se obtiene del LDAP. A dicho usuario se le asigna un **rol** según la función que este ciudadano cumpla en el

---

<sup>1</sup> LDAP (Lightweight Directory Access Protocol): Protocolo Ligero de Acceso a Directorios.

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

sistema y para obtener el acceso pleno a las acciones que realiza, se le asignan los **permisos** correspondientes a dicho rol. Los ciudadanos pertenecen a una o varias **oficinas** según el caso, donde les corresponde uno o varios **puestos de trabajo** desde donde podrán acceder al sistema.

### 2.3 Concepción de los módulos del sistema

La Plataforma de Identificación de la UCI está conformada por un conjunto de 7 módulos. Estos son: Gestión de citas, Captación de datos, Captación de imágenes, Supervisión, Personalización, Entrega y Reportes. (Ver figura 2) Los elementos de configuración y seguridad de cada uno de ellos serán gestionados y configurados desde el módulo de administración por medio de un usuario con roles y/o permisos de administrador del sistema, garantizando de esta manera la seguridad del mismo.

Los cambios que se realicen desde la administración afectarán a cada módulo de forma general y/o particular.

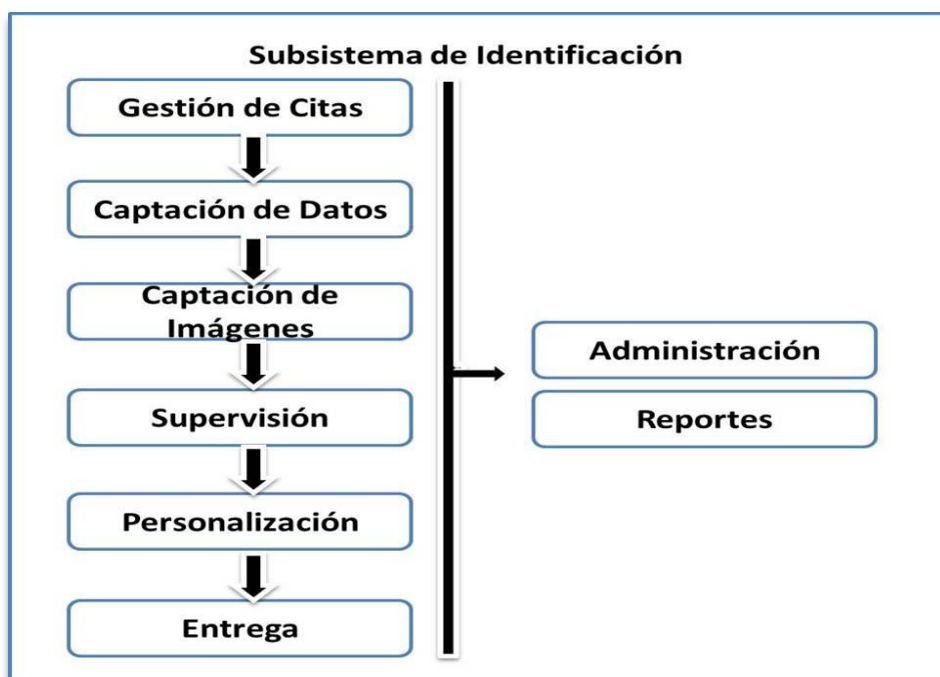


Figura 2 Concepción de los módulos.

## 2.4 Especificación de los Requerimientos del Software

En vista de tener un mejor entendimiento de los requerimientos que el sistema debe cumplir o poseer además de las condiciones que necesita el usuario para hacer uso de la misma, se confeccionó un catálogo de requisitos, donde se especifican los requisitos funcionales (RF) y los requisitos no funcionales (RNF).

### 2.4.1 Requisitos Funcionales del Sistema

Los requisitos funcionales especifican acciones o servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante diferentes situaciones. Es decir, especifican el comportamiento de entrada y salida del sistema. Se identificaron los siguientes requisitos funcionales:

#### **RF1: Autenticar Usuario**

RF1.1: Introducir usuario y contraseña

RF1.2: Permitir acceso al sistema

RF1.3: Denegar acceso al Sistema

#### **RF2: Gestionar Ciudadano**

RF2.1: Buscar ciudadano (ci, solapín)

RF2.2: Mostrar ciudadano (nombre (s), apellidos, ci, sexo)

RF2.3: Adicionar ciudadano al sistema

#### **RF3: Gestionar Usuarios**

RF3.1: Crear usuario. (nombre, correo, estado)

RF3.2: Mostrar usuario

RF3.3: Modificar estado de usuario

#### **RF4: Gestionar Rol**

RF4.1: Crear rol (nombre, descripción, estado)



## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

RF4.2: Mostrar rol

RF4.3: Modificar estado del rol

### **RF5: Asignar Permiso a Rol**

RF5.1: Buscar por usuario

RF5.2: Asignar permiso

RF5.3: Eliminar Permiso

### **RF6: Asignar Rol a Usuario**

RF6.1: Asignar rol

RF6.2: Eliminar asignación

### **RF7: Gestionar Oficinas**

RF7.1: Crear oficinas (nombre, estado, cantidad de puestos de trabajo)

RF7.2: Mostrar oficinas

RF7.3: Modificar oficinas (estado, cantidad de puestos de trabajo)

### **RF8: Gestionar Puestos de Trabajo**

RF8.1: Crear puestos de trabajo (nombre, estado, MAC, tipo)

RF8.1.2: Asignar puesto de trabajo (usuario, oficina)

RF8.2: Mostrar puestos de trabajo

RF8.3: Modificar estado de los puestos de trabajo

### **RF9: Configurar Aplicación**

RF9.1: Configurar módulos

RF9.2: Configurar tema de interfaz

### **RF10: Registrar Operaciones**

RF10.1: Filtrar operaciones (fecha, acción, usuario)

RF10.2: Mostrar operaciones

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

### **RF11: Editar Perfil del Usuario del Sistema**

RF11.1: Cambiar Contraseña de Usuario del Sistema

### **RF 12: Configurar Fuente Externa**

#### **1.4.2 Descripción de los Requisitos Funcionales**

Luego de ser identificados los requisitos funcionales, se realiza una descripción de los mismos con el objetivo de facilitar la implementación del sistema ya que en ella se definen las entidades que se van a utilizar, las condiciones que debe cumplir la aplicación inicialmente, explica la secuencia de actividades para llevar a cabo las funcionalidades descritas, así como el resultado que se espera después de realizada cada una de las operaciones. A continuación se muestra la descripción del requisito funcional "Gestionar Rol" (ver tabla 2). Para un análisis profundo de las especificaciones de los requisitos funcionales ver ANEXO 2: "Descripción de las Funcionalidades".

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador.
Funcionalidades tratadas	RF4, RF4.1, RF4.2, RF4,3
Conceptos tratados	Rol, Ciudadano, Usuario
Descripción básica	<p>1 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"><li>• Gestionar Ciudadano</li><li>• Gestionar Usuario</li><li>• Gestionar Oficina</li><li>• Gestionar Puestos</li><li>• Gestionar Rol</li><li>• Asignar Rol a Usuario</li><li>• Asignar Permisos</li><li>• Eliminar Permisos</li><li>• Nomencladores</li><li>• Registros de Operaciones</li><li>• Configurar Aplicación</li><li>• Logs</li></ul>

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

2 Si el usuario selecciona la opción "Gestionar Rol" el sistema muestra la interfaz "Gestionar Rol" (ver figura 1) donde se visualizará un listado con los roles que se han añadido al sistema, su estado y una descripción de los mismos. En caso de no existir ningún rol, el listado se mostrará vacío, con un mensaje "No hay rol, Adicione roles al sistema".

**Acciones:**

- Adicionar
- Modificar
- Mostrar información

2.1 Si el usuario selecciona la opción "Adicionar" se mostrará la interfaz "Adicionar Rol" (ver figura 2) con los siguientes parámetros a llenar:

**Parámetros a llenar:**

- Nombre
- Descripción
- Estado

**Acciones:**

- Salvar
- Cancelar

2.1.2 Si se selecciona la opción "Salvar" se adiciona el rol a la lista de roles y a la base de datos del sistema. Se cerrará la ventana "Añadir Rol" y se mostrarán los datos del rol añadido en la lista de roles.

2.1.3 Si se selecciona la opción "Cancelar" se cerrará la ventana "Añadir Rol".

2.2 Si el usuario selecciona la opción "Modificar" se mostrará la interfaz "Modificar Rol" (ver figura 3) con los siguientes parámetros a llenar:

**Parámetros a llenar:**

- Descripción
- Estado

**Acciones:**

- Salvar
- Cancelar

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

2.2.1 Si se da clic en el botón "Salvar" se actualizan en el listado de roles los datos modificados y en la base de datos del sistema. Se cerrará la ventana "Modificar Rol".

2.2.3 Si se da clic en el botón "Cancelar" se cerrará la ventana "Modificar Rol".

2.3 Si el usuario selecciona la opción "Mostrar información" se desplegará del rol correspondiente el ciudadano, usuario y estado correspondiente a dicho rol. (Ver figura 4)

### Prototipos

Figura 1. Gestionar Rol:

The screenshot displays a web application interface for role management. The header includes the logo 'ADMINISTRACIÓN' with the tagline 'Gestione y Configure a su Gasto' and the text 'SISTEMA DE IDENTIFICACIÓN'. A user greeting 'Bienvenido: Leonardo Díaz Novo' is visible in the top right. The left sidebar contains a menu with options like 'Administración', 'Gestionar Ciudadano', 'Gestionar Usuario', 'Gestionar Oficina', 'Gestionar Puestos', 'Gestionar Rol', 'Asignar Rol a Usuario', 'Asignar Permisos', 'Eliminar Permisos', 'Nomencladores', 'Registro Operaciones', 'Configurar Aplicación', 'Logs', 'Captación', 'Supervisión', and 'Entrega'. The main content area is titled 'Gestionar Rol' and features a table titled 'Listado de Roles'.

Nombre	Descripción	Estado
Administrador	Encargado de administrar la aplicación	Activo
Captación	Encargado de captar los datos e imagenes	Activo
Entrega	Encargado de gestionar la entrega de los documentos de identificación	Activo
Supervisor	Encargado de la supervisión de datos	Activo

1 Mostrando 1 a 4 de 4 registros + Adicionar

UCI, viernes, 14 de junio de 2013

Figura 2. Adicionar Rol:

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

ADMINISTRACIÓN  
Gestione y Configure a su Gusto

SISTEMA DE IDENTIFICACIÓN Bienvenido: Leonardo Díaz Novo

Gestionar Rol

Listado de Roles

Nombre
Administrador
Captación
Entrega
Supervisor

1 Mostrando 1 a 4 de 4 registros

+ Adicionar

+ Adicionar

Nombre: Administrador

Descripción:

Estado:  Activo

Salvar Cancelar

UCI, viernes, 14 de junio de 2013

Figura 3. Modificar Rol:

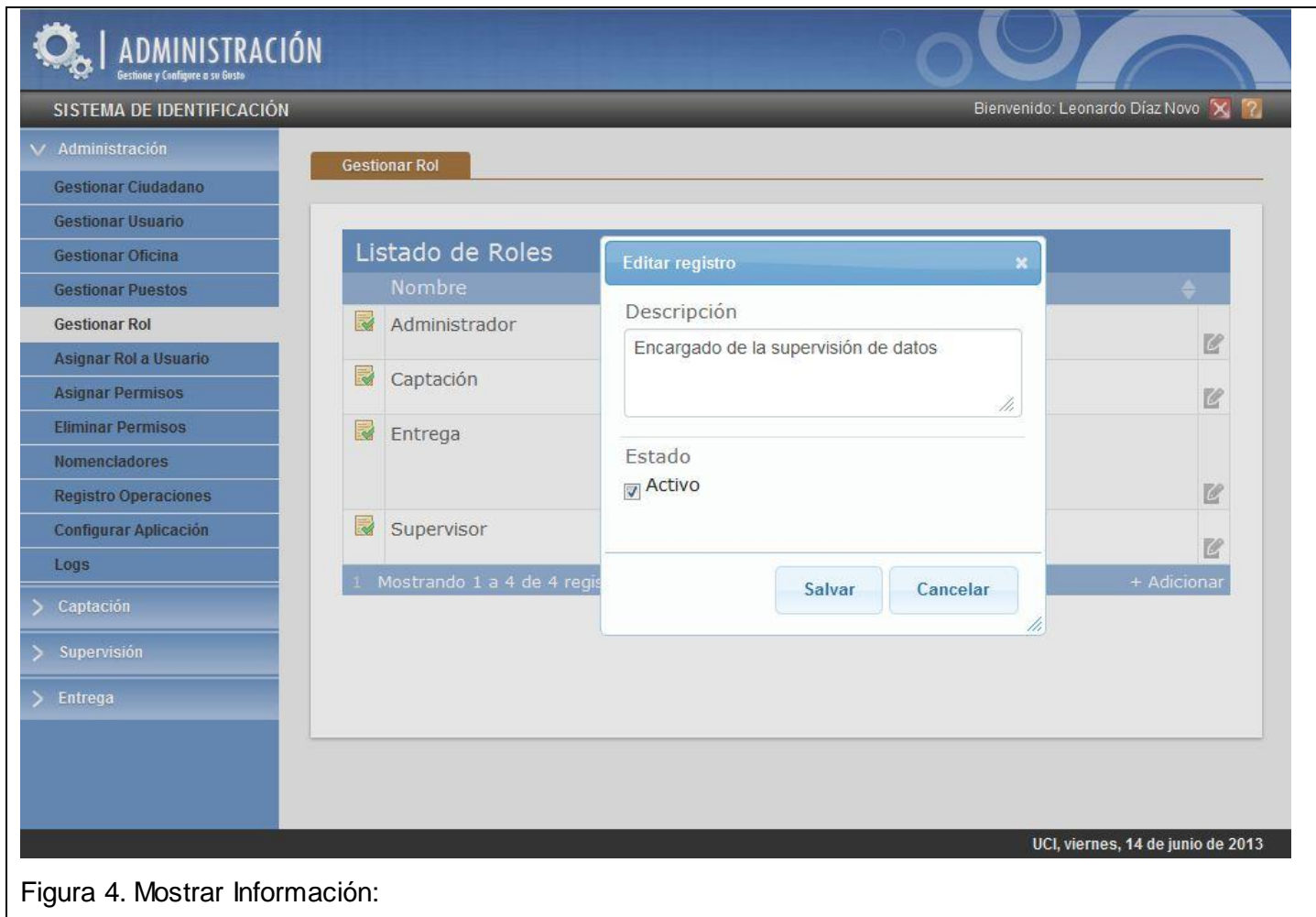


Figura 4. Mostrar Información:



**ADMINISTRACIÓN**  
Gestione y Configure a su Gasto

SISTEMA DE IDENTIFICACIÓN Bienvenido: Leonardo Díaz Novo

**Gestionar Rol**

**Listado de Roles**

Nombre	Descripción	Estado
Administrador	Encargado de administrar la aplicación	Activo
Captación	Encargado de captar los datos e imagenes	Activo
Entrega	Encargado de gestionar la entrega de los documentos de identificación	Activo

**Usuarios que son de Rol - Entrega**

Nombre de la Persona	Usuario	Estado del Usuario
Asiel Echemendia Carrasco	aecarrasco	Activo
Supervisor	Encargado de la supervisión de datos	Activo

1 Mostrando 1 a 4 de 4 registros + Adicionar

UCI, sábado, 15 de junio de 2013

Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Se adicionarán los roles al sistema.

**Tabla 2 Descripción de la funcionalidad "Gestionar Rol"**

### 2.4.2 Requisitos No Funcionales del Sistema

Los requisitos no funcionales especifican los criterios que se deben usar para el funcionamiento de un sistema, en lugar de un comportamiento específico. Mientras que los requisitos funcionales definen lo que el sistema debería de hacer, los requisitos no funcionales verifican cómo un sistema debería ser.

#### Requisitos de Usabilidad

RnF1. El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

- Conocimientos básicos relativos al uso de una computadora.

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

- b) Conocimientos básicos del sistema operativo *Windows*.
- c) Conocimientos sólidos relativos de administración de un sistema.

**RnF2.** El sistema será distribuido en idioma español.

**RnF3.** Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.

**RnF4.** El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación.

- a) Menús laterales y desplegados que permitan el acceso rápido a la información.
- b) Menú de soporte que facilite el acceso a herramientas utilitarias, notificaciones del sistema y ayuda integrada.

### **Requisitos de Fiabilidad**

**RnF5.** El sistema debe estar disponible las 24 horas del día los 7 días de la semana realizándose copias de seguridad semanalmente tanto de la aplicación como de la base de datos.

**RnF6.** El tiempo medio entre fallos no debe ser superior a 1 hora.

**RnF7.** El sistema estará 1 hora fuera de operación luego de haber fallado 3 veces.

### **Requisitos de Eficiencia**

**RnF8.** El sistema debe ser capaz de soportar una cantidad escalable de peticiones y dar respuestas efectivas y rápidas.

**RnF9.** El sistema debe hacer un uso eficiente de los recursos de *hardware* donde se encuentre desplegado.

### **Requisitos de Soporte**

**RnF10.** El soporte de las aplicaciones y componentes utilizados en el desarrollo del sistema será responsabilidad de los proveedores de los mismos.

**RnF11.** El sistema dispondrá de un año de soporte técnico donde se corregirán las fallas que atenten contra el buen funcionamiento del sistema.



## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

### Requisitos de Restricciones de diseño

- RnF12. Plataforma de desarrollo.NET 4.0 utilizando *Visual Studio Ultimate 2010*.
- RnF13. Biblioteca jQuery 1.9.
- RnF14. Se utilizará ASP.NET *Web Forms* 4.0
- RnF15. Para el acceso a datos se utilizara el ORM NHibernate 3.1.0.

### Requisitos para la documentación de usuarios en línea y ayuda del sistema.

- RnF16. El sistema deberá contar con un manual de usuario y una ayuda del sistema para facilitar el trabajo de los usuarios con el sistema.

### Requisitos de Interfaz

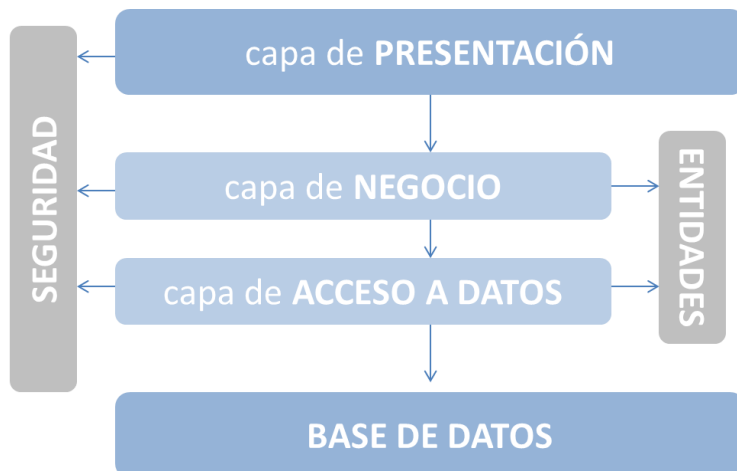
- RnF17. El sistema dispondrá de un diseño ameno e intuitivo acorde las características de la institución.

## 2.5 Arquitectura de la Solución

La arquitectura de un *software* es determinante para su éxito pues se encarga de establecer los fundamentos para que los miembros del equipo de desarrollo en sus diferentes roles se guíen por una línea de trabajo común y así alcanzar los objetivos de manera exitosa.

La arquitectura del módulo de administración de la Plataforma de Identificación de la UCI está basada en la arquitectura n-capas la cual brinda escalabilidad, flexibilidad y mejoras en las posibilidades de mantenimiento, debido a que cada capa es independiente de la otra, los cambios o actualización pueden ser realizados sin afectar la aplicación como un todo. (Ver figura 3)

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.



*Figura 3 Vista lógica de la arquitectura de software*

**Capa de Presentación:** Es la capa con la que interactúa el usuario, las pantallas que se le muestra para que interactúe con el programa, comunicándole la información y recolectando la información suministrada por el usuario en un mínimo de proceso (realiza validaciones para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio intercambiando los datos o registros necesarios, es la interfaz gráfica del programa y debe ser lo más amena posible para una mejor comunicación con el usuario.

**Capa de negocio:** Se denomina capa de negocio o lógica del negocio porque es aquí donde se encuentran todas las funcionalidades que dan respuesta a los requisitos del sistema. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

**Capa de acceso a datos:** Esta capa se utiliza con el objetivo de estandarizar el acceso a datos, y eliminar el acoplamiento de los componentes de negocio al gestor de base de datos. De esta forma se puede minimizar el impacto en caso de ocurrir cambios en el modelo de datos. Es la única capacitada para interactuar con la capa de base de datos, y a su vez expone servicios a la capa de negocio.

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

**Capa de base de datos:** Está constituida por todo el conjunto de tablas y procedimientos que permiten el almacenamiento de la información recolectada y procesada. La única capa encargada de interactuar con ésta es la capa de acceso a datos.

### 2.6 Diagrama de Clases del Diseño

El diagrama de clases pertenece al artefacto Modelo de Diseño, empleado para representar y documentar el diseño. Sirve para visualizar las relaciones entre las clases que involucran el sistema. Un diagrama de clases está compuesto por los siguientes elementos: Clase (atributos, métodos y visibilidad) y Relaciones (Herencia, Composición, Agregación, Asociación y Uso).

Se modelaron los diagramas de clases del diseño con estereotipos *web* mostrando los tipos de clases utilizadas, la relación que tendrán al momento de la ejecución de la aplicación y cómo será la comunicación entre estas. A continuación se muestra el diagrama de clases para la funcionalidad "Gestionar Rol", (ver figura 4) el resto de los diagramas se muestran en el ANEXO 3: "Diagrama de Clases del Diseño".

# :CAPÍTULO II: PROPUESTA DE SOLUCIÓN:

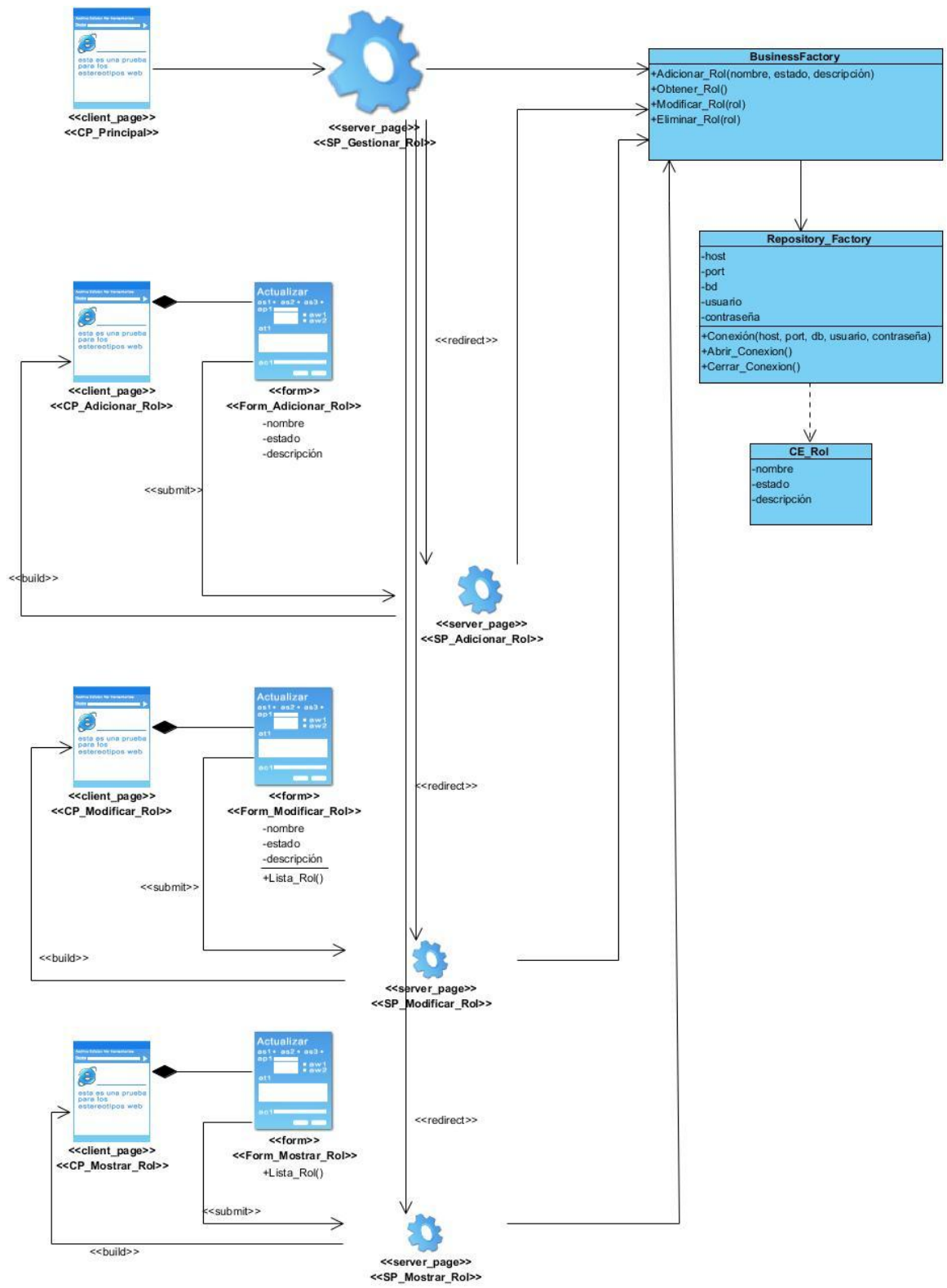


Figura 4 Diagrama de clases "Gestionar Rol"

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

A continuación se describen de manera general las clases del diagrama de clases "Gestionar Usuarios" (ver tabla 3).

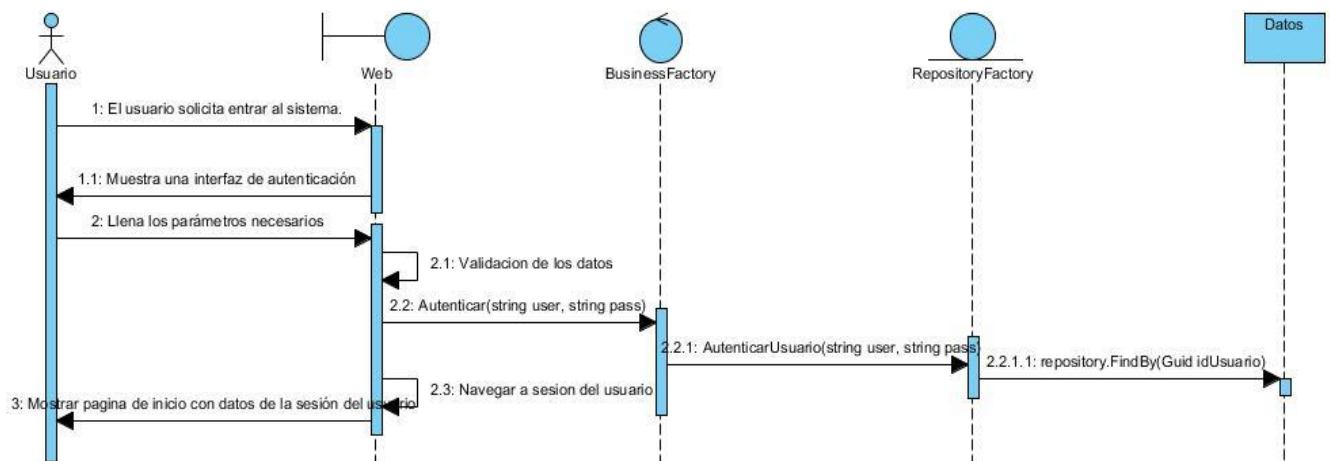
Nombre	Tipo	Descripción
BusinessFactory	Lógica del negocio	Es la encargada de gestionar las operaciones referentes a la gestión de roles.
RepositoryFactory	Acceso a Datos	Es la encargada de gestionar las operaciones fundamentales que se ejecutan en la base de datos.
CE_Rol	Entidad	Entidad que contiene los datos e información de los roles del sistema.

*Tabla 3 Descripción de las clases del diagrama de clases "Gestionar Rol"*

### 2.7 Diagrama de Secuencia

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema y muestran un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Entre los diagramas de interacción se encuentran los diagramas de secuencia.

Los diagramas de secuencia reflejan cómo se comunican los objetos en una interacción a través de los mensajes intercambiados entre las clases. A continuación se muestra el diagrama de secuencia para la funcionalidad "Autenticar Usuario" (ver figura 5). El resto de los diagramas se encuentran en el ANEXO 5.



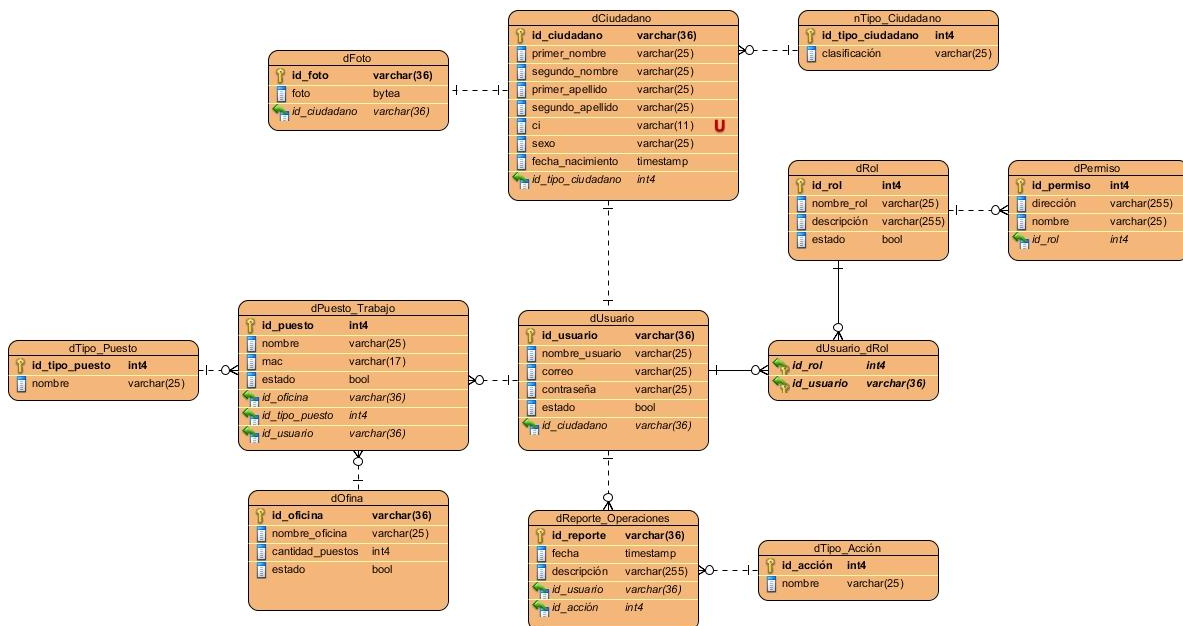
*Figura 5 Diagrama de secuencia "Autenticar Usuario"*

## 2.8 Diseño de la Base de Datos

Una de las tareas más importantes en el desarrollo de un sistema de gestión es el diseño de la persistencia de los datos que maneja. Se diseña con el fin de acceder a los datos cuando es necesario. Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Un diseño correcto es esencial para lograr los objetivos fijados ya que posibilita mayor calidad en la implementación del sistema. (27)

### 2.8.1 Modelo de Datos

Un modelo de datos describe la representación lógica y física de los datos persistentes usados por el sistema. A continuación se muestra en la figura 6 el modelo de datos del sistema:



**Figura 6 Modelo de datos**

### 2.8.2 Descripción del Modelo de Datos

Para una mejor comprensión del modelo de datos se describe las tablas que lo componen (ver Tabla 4). Se realizó además una descripción específica para cada una de las entidades. Las mismas se encuentran en el ANEXO 4: “Descripción de las entidades del Modelo de Datos”.

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

Nombre de la Tabla	Tipo	Descripción
dCiudadano	Tabla de datos	Almacena los datos de las personas existentes en la base de datos y las que se crean en el propio sistema.
nTipo_Ciudadano	Tabla de nomenclador	Registra los tipos de ciudadanos.
dUsuario	Tabla de datos	Registra los datos de los usuarios que acceden al sistema.
dRol	Tabla de datos	Contiene los roles que pueden tener un usuario del sistema.
dUsuario_dRol	Tabla de datos	Representa la relación existente entre la tabla dUsuario y la tabla dRol.
dPermisos	Tabla de datos	Contiene los permisos que se le asignan a los roles.
dOficina	Tabla de datos	Contiene las oficinas que hacen uso del sistema.
dPuesto_Trabajo	Tabla de datos	Almacena los puestos de trabajo de cada oficina.
nTipo_Puesto	Tabla de nomenclador	Nomenclador para los tipos de puestos que pueden existir.
dReporte_Operaciones	Tabla de datos	Almacena las operaciones realizadas por los usuarios.
nTipo_Acción	Tabla de nomenclador	Nomenclador para los tipos de acciones que pueden realizar los usuarios.

*Tabla 4 Composición de tablas del modelo de datos*



## 2.9 Patrones de Diseño

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. A continuación se muestran los patrones utilizados en el desarrollo de la solución propuesta.

### **Creador:**

La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. Puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilización. Guía la asignación de responsabilidades relacionadas con la creación de objetos. El patrón creador pertenece a los patrones GRASP<sup>1</sup>. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.(28) En la figura 7 se muestra un ejemplo de su uso.

```
public class Email
{
    private MailMessage Mensaje;
    private SmtClient cliente;

    public Email(string server,string usser,string password,int port)
    {
        Mensaje = new MailMessage();
        cliente = new SmtClient();
        this.Usser = usser;
        this.Password = password;
        this.Server = server;
        this.Port = port;
    }
}
```

*Figura 7 Ejemplo del uso del patrón creador*

### **Singleton (Instancia Única):**

Se encuentra dentro de los patrones creacionales pertenecientes a los GOF<sup>2</sup>. Se asegura que solo se pueda crear una instancia de la clase y ofrece un punto global de acceso a esta instancia. El uso del patrón *singleton* permite que los servicios puedan ser creados solo una vez.(29) En la figura 8 se muestra un ejemplo del uso de este patrón.

<sup>1</sup> GRASP(*General Responsibility Assignment Software Patterns*): Grupo de patrones de diseño.

<sup>2</sup> GOF(*Gang Of For*): Grupo de patrones de diseño.



## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

```
public class BusinessFactory
{
    private static BusinessFactory instance = null;
    public static BusinessFactory Instance
    {
        get
        {
            if (BusinessFactory.instance == null)
                BusinessFactory.instance = new BusinessFactory();

            return BusinessFactory.instance;
        }
    }
}
```

*Figura 8 Ejemplo del uso del patrón singleton*

### 2.10 Elementos de Seguridad en el Sistema

La seguridad en las aplicaciones *web* son de vital importancia, por lo que se debe comprender e implementar las mismas para protegerlas. ASP.NET funciona junto con Microsoft .NET Framework para ayudar a proporcionar aplicaciones *web* seguras. Para dar seguridad al sistema, se hizo uso de algunos elementos de seguridad entre los que se encuentran los siguientes:

#### **Autenticación**

La autenticación consiste en verificar que el usuario que trata de identificarse es válido, usualmente se implementa con una contraseña en el momento de iniciar una sesión. Entre los tipos de técnicas que existen para permitir realizar la autenticación de un usuario, se eligió "algo que el individuo solamente conoce"; en este caso una contraseña. La primera vez que un usuario perteneciente al sistema se autentique, deberá usar su contraseña del dominio LDAP de la UCI, pues ha sido registrado al sistema con dicha contraseña. Una vez autenticado se le brinda la opción de editar su perfil, cambiando su contraseña por otra que le permita entrar por el dominio local. Luego de esto, ya no podrá acceder más por la contraseña del dominio LDAP.

#### **Autorización**

La autorización sucede después de la autenticación. Esta limita los derechos de acceso mediante la aprobación o negación de permisos específicos a un usuario autenticado. La autorización es el proceso de determinar si un usuario tiene los permisos para el acceso a los recursos. Para esto

## .:CAPÍTULO II: PROPUESTA DE SOLUCIÓN:.

el sistema utilizó el modelo de control de acceso basado en roles RBAC<sup>1</sup>. De esta manera los permisos son asignados a los roles, lo cual permite que los mismos tengan acceso a los módulos necesarios y suficientes para efectuar sus responsabilidades. Un usuario con rol de administrador, poseerá todos los permisos sobre el sistema, y tendrá acceso pleno a todos los módulos existentes en el mismo. El sistema permite además la asignación de permisos especiales, donde el administrador podrá otorgar permisos a usuarios determinados, a acceder a partes de la aplicación, distintas a las que le pertenecen según su rol.

### **Auditoría**

Es el control de eventos y registro de acciones que ocurren en el sistema, realizado con el fin de verificar si sus actividades se desarrollan adecuadamente, de acuerdo con las normas informáticas. Para chequear el correcto comportamiento del sistema, el mismo mostrará un registro de las operaciones realizadas en el sistema a través de un filtrado por fecha, usuario y acción que se desee verificar.

### **2.11 Conclusiones del Capítulo**

La modelación del negocio a través de un modelo de dominio sirvió para relacionar todos los conceptos del negocio dejando claro los elementos que intervienen en el funcionamiento del mismo. Los requisitos funcionales y no funcionales presentados dejaron ver las características y necesidades del módulo de administración. Con la arquitectura definida se obtendrá una aplicación robusta y segura debido a las particularidades de la misma. Al realizarse el diagrama de clases del diseño quedó representada la estructura de clases y métodos que serán implementados en la fase de implementación, así como los diagramas de secuencia, los cuales mostraron los aspectos dinámicos de un sistema. Se definió el modelo de datos, donde se identificaron las entidades persistentes que contienen la información que se procesa, especificándose sus atributos y las relaciones entre ellas. El uso de los patrones ayudó a obtener un diseño flexible y reusable.

---

<sup>1</sup> RBAC (Role-Based Access Control): Modelo de control de acceso.

## CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA

### 3.1 Introducción

El presente capítulo tiene como objetivo abordar la implementación en base a las funcionalidades, donde se describe cómo se implementan en términos de componentes los elementos del modelo de diseño. Se modelan los diagramas de componentes y de despliegue, dando una visión de la estructura lógica y física del sistema. Además se especifica el tipo de prueba realizada a todas las funcionalidades necesarias.

### 3.2 Implementación

Durante la implementación del sistema se define la organización del código, se implementan los elementos del diseño, se integran los resultados en un sistema ejecutable y se distribuyen físicamente asignando componentes ejecutables a nodos. El objetivo de esta actividad es programar la solución, siguiendo la estructura de componentes propuesta en actividades anteriores.

#### 3.2.1 Diagrama de Despliegue

El modelo de despliegue es utilizado para capturar los elementos de *hardware* y las conexiones entre ellos. Se utiliza para visualizar la distribución de los componentes de *software* en los nodos físicos. A continuación se muestra en la figura 9 el diagrama de despliegue del sistema:

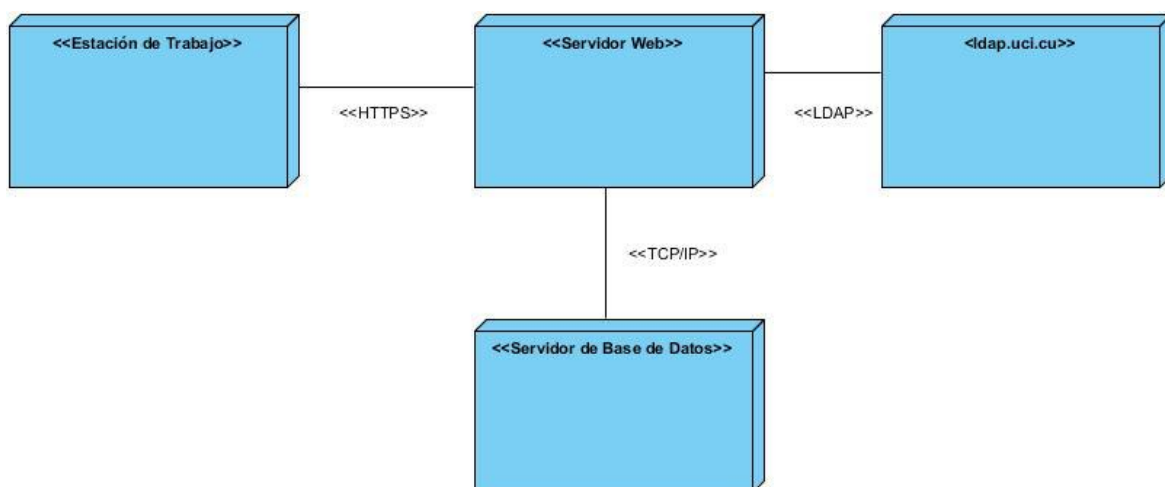


Figura 9 Diagrama de despliegue

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

**Estación de trabajo:** Se conecta al servidor *web* a través del protocolo HTTPS para visualizar la aplicación. Esta PC<sup>1</sup> tendrá instalado Microsoft .NET Framework 4.0.

**Servidor Web:** Contiene la aplicación a la cual se accederá desde las estaciones de trabajo. Tendrá instalado Microsoft .Net Framework v4.0.

**Servidor de Base de Datos:** Contiene la base de datos donde se encuentran registradas los datos pertinentes a los ciudadanos añadidos en el sistema con sus datos. Deberá poseer como gestor de base de datos PostgreSQL 9.1. La comunicación desde el servidor *web* y este servidor de bases de datos será a través de TCP/IP.

**Servidor LDAP de la UCI:** Es el servidor que utiliza la universidad para gestionar los usuarios de manera centralizada. La aplicación *web* contenida en el servidor *web* accede a este para obtener la información de los ciudadanos y almacenarlas en la base de datos del sistema. El protocolo utilizado para la conexión a este servidor desde el servidor *web* será LDAP.

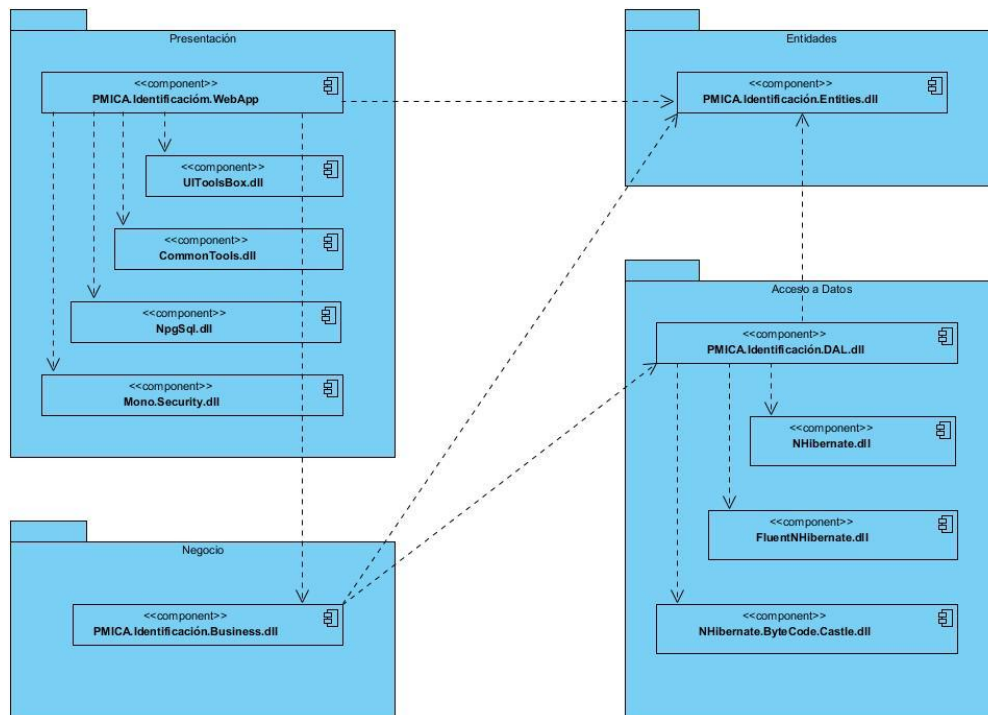
### 3.2.2 Diagrama de Componentes

El diagrama de componente describe como queda estructurado el sistema mediante elementos físicos y sus relaciones entre componentes de *software*, Estos pueden ser de código fuente, binarios, ejecutables entre otros. Los componentes se agruparan en paquetes según la estructura lógica del sistema que mediante capas bien definidas facilitará la implementación del sistema. En la figura 10 se muestra el diagrama de componentes correspondiente al módulo de administración, el cual se modeló teniendo en cuenta las distintas capas del sistema, agrupándose estas en paquetes que contendrán los distintos componentes que se utilizan para el correcto funcionamiento del sistema.

---

<sup>1</sup> PC (Personal Computer): Computadora Personal.

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.



**Figura 10 Diagrama de Componentes**

El paquete de presentación está compuesto por varios componentes que tienen como finalidad mostrarle las interfaces para su interacción con el sistema al usuario y las configuraciones generales de la aplicación *web*. En este paquete se hace uso del ensamblado **UIToolBox.dll** componente desarrollado por el proyecto Identidad Cuba que contiene un conjunto de controles *web* para ayudar a los desarrolladores en la creación de las interfaces *web*. También utiliza los ensamblados **Npgsql.dll** y **Mono.Security.dll** que harán función de driver para la conexión con la base de datos y deben estar en el proyecto principal de la aplicación, además de **CommonTools.dll**, ensamblado para las configuraciones. En el paquete negocio el componente **PMICA.Identificación.Business.dll** agrupa todas las clases que contenga la lógica del sistema, este componente a su vez hace uso del componente **PMICA.Identificación.Entities.dll**, ensamblado perteneciente al paquete entidades y que agrupa las entidades del sistema, también utiliza el componente **PMICA.Identificación.DAL.dll** ensamblado que agrupa las principales clases para el acceso a datos y las entidades persistentes del sistema que mediante el uso de los componentes **NHibernate.dll**, **FluentNHibernate.dll** y **NHibernate.ByteCode.Castle.dll** facilitaran la conversión de datos entre el sistema de tipos utilizado en el lenguaje de

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

programación orientado a objetos y el de la base de datos relacional; todos estos componentes estarán agrupados en el paquete acceso a datos.

### 3.2.3 Estándares de Codificación

Para una mejor organización del código se establecieron los estándares de codificación y se definieron las convenciones a ser utilizadas. (Ver información más ampliada en el documento del proyecto: *PMICA\_Estándares de codificación para C# 1.0*) A continuación se exponen las convenciones Camel y Pascal, explicando en qué consisten y en qué momento se hará uso de ellas.

**Convención Pascal:** El primer carácter de cada palabra es en mayúscula y el resto en minúscula.

**Convención Camel:** El primer carácter de cada palabra es en mayúscula (excepto la primera palabra) y el resto en minúscula.

Uso	Convención	
	Camel	Pascal
Para el nombre de las clases	X	
Para el nombre de las interfaces	X	
Para el nombre de los métodos		X
Para el nombre de variables y parámetros	X	
Para el nombre de los ficheros		X

*Tabla 5 Uso de las convenciones Camel y Pascal*

#### **Variables (según la convención Camel):**

- Para el nombre de las variables se usará el significado de las palabras, evitando las abreviaturas para un mejor entendimiento.

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

- Evitar el uso de caracteres simples para el nombre de las variables *i*, *n*, *s* etc, con excepción de las variables de los ciclos. En lugar de esto, serán usados nombres como *index*, *temp*, etc.
- Los nombres de las variables locales no llevarán *underscores* (`_`).
- Todas las variables miembros de las clases serán prefijadas con *underscore* (`_`) para ser diferenciadas de otras variables.
- Evitar usar nombres de variables que coincidan con palabras reservadas.

### Indentación y espaciado:

- Usar TAB para la indentación. Definir el tamaño del Tab como 4.
- No usar SPACES.

### Buenas prácticas de programación:

- Cada método debe cumplir solamente una función. No se deben combinar más de una función por método, tratar además que las funcionalidades sean los más atómicas posibles.
- Siempre vigilar parámetros no esperados. Por ejemplo, si se usa un parámetro con dos posibles valores, no asumir que si no es el primero sea el segundo, comprobar efectivamente cada caso.
- No usar directamente números o cadenas como constantes en el código. Declarar las constantes en la parte superior de los ficheros.
- Se debe evitar declarar constantes en todos los ficheros es recomendable que exista un fichero *config* o una base de datos para las mismas.
- Convertir las cadenas a mayúsculas o minúsculas antes de compararlas.
- Usar *String.Empty* en lugar de `""`.
- Usar *enum* siempre que sea requerido. No usar números o cadenas para indicar valores discretos.
- No programar más de una clase por cada fichero.

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

- Evitar tener ficheros muy largos. Se deben dividir en dos o más clases.
- Evitar el paso de demasiados parámetros a los métodos, tratar de que no haya métodos de más de 3 parámetros.
- Especificar en el *AssemblyInfo file* la información del número de versión, descripción, desarrollador.
- Organizar los ficheros en carpetas apropiadas. Usar dos niveles de profundidad, no más de 10 carpetas en la raíz y no más de 5 subcarpetas en cada carpeta hija.
- Todas las conexiones a bases de datos, *sockets*, ficheros etc, siempre son cerradas en el bloque *finally*.

### 3.2.4 Tratamiento de Errores

Para garantizar una mayor integridad y confidencialidad en los datos que utiliza el sistema se adoptan las siguientes estrategias para el tratamiento de errores:

- Los datos introducidos se validan verificando que los mismos se correspondan con el formato del campo.
- Cuando el usuario introduzca la información incorrecta o deje campos vacíos se mostrará un mensaje en color rojo (ver figura 11), o una ventana en frente (ver figura 12) indicando el error.
- Los mensajes de error que emitirá el sistema se mostrarán en un lenguaje de fácil comprensión para los usuarios.
- El tratamiento de las excepciones se realiza a través de las sintaxis `try {...} catch (Exception ex) {...}`.

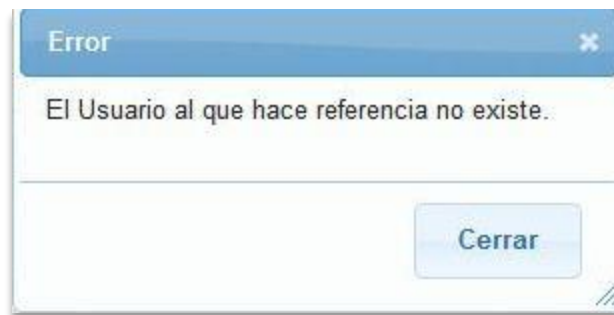
A continuación se muestra un ejemplo del tratamiento de errores en el sistema.





A screenshot of a web form. At the top, there is a text input field containing the text "777777hh". To the right of the input field is a blue button labeled "Buscar". Below the input field, there is a red error message that reads "Solo puede entrar números.".

*Figura 11 Respuesta del sistema al entrar letras en el campo de ci del ciudadano*



*Figura 12 Respuesta de sistema al asignar un puesto de trabajo a un usuario que no existe*

### 3.3 Pruebas de Software

En cada sistema informático es de vital importancia la calidad con la que son desarrollados los mismos. Un buen sistema (uno de alta calidad) es aquél que cumple con las necesidades del cliente demostrando ser útil, confiable y flexible. Las pruebas de *software* constituyen un factor relevante para verificar la calidad de un producto de *software*, representando una inspección final de las especificaciones del diseño y la implementación, donde se identificarán posibles fallos. Una prueba no es más que una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones específicas, los resultados son observados y registrados. Las pruebas de *software* son un elemento crítico para la garantía de calidad del *software* y representan una revisión final de las especificaciones, del diseño y de la codificación. Los objetivos principales de realizar una prueba son:

- ✓ Detectar un error.
- ✓ Tener un buen caso de prueba, es decir que tenga más probabilidad de mostrar un error no descubierto antes.

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

- ✓ Descubrir un error no descubierto antes (éxito de la prueba).

Un producto puede probarse siguiendo dos criterios:

- ✓ Conocimiento del funcionamiento del producto (caja blanca).
- ✓ El conocimiento de la función específica para la que fue diseñado el producto (caja negra).

En búsqueda del aseguramiento de la calidad del módulo de administración, se decide realizar al *software* pruebas de caja blanca (usando la técnica de pruebas unitarias) y pruebas de caja negra.

### 3.3.1 Pruebas de Caja Blanca (estructural)

Las pruebas de caja blanca son aquellas que se llevan a cabo sobre las funcionalidades de manera interna. Se desarrolla con el fin de asegurar que todas las piezas del sistema tienen una operación interna que se ajusta a las especificaciones y que todos sus componentes internos se han aprobado en forma adecuada.

#### Pruebas Unitarias

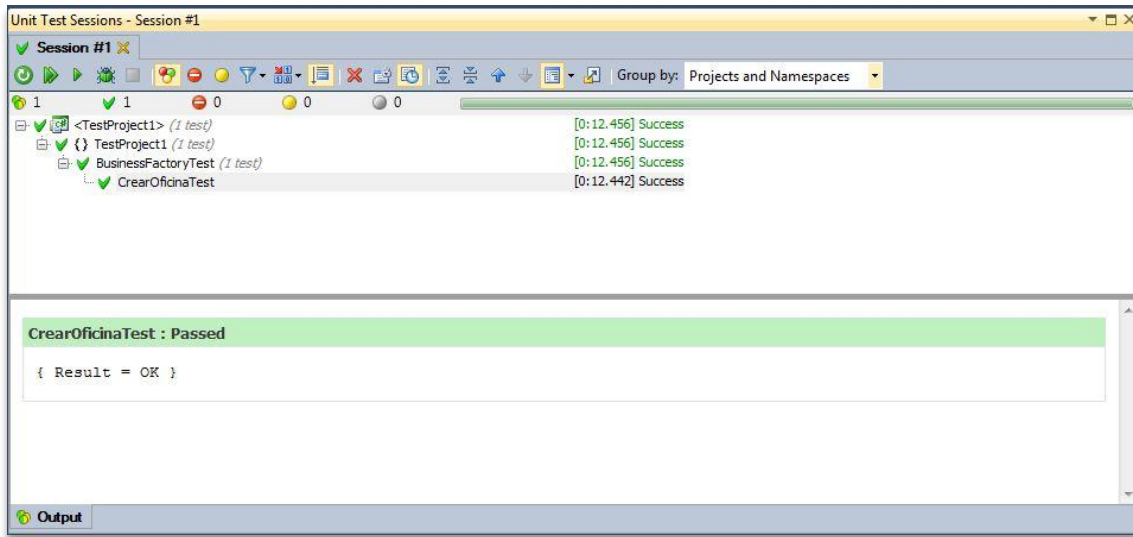
Las pruebas de unidad son la primera fase de las pruebas que se le aplican a cada módulo de un *software* de manera independiente. Su objetivo es verificar que el módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. Consiste en ejecutar un código fuente llamando directamente a los métodos de una clase, pasándole a estos los parámetros apropiados. Con las pruebas unitarias se consigue el aislamiento de las partes del código y la demostración de que estas partes no contienen errores. Es por ello que se consideran a las pruebas unitarias como uno de los tipos de pruebas más importantes que se le aplican a los *softwares*, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Con el fin de lograr el correcto funcionamiento de las funcionalidades se realizaron las pruebas necesarias para cada iteración de desarrollo. Durante la ejecución de las pruebas se creó para cada clase un caso de prueba con los métodos necesarios para verificar su ejecución. A continuación se muestra un ejemplo del caso de prueba creado para la clase *BusinessFactory*.

Se creó la clase *BusinessFactoryTest* donde se implementa el método de prueba

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

*CrearOficinaTest()*, el cual verifica que los datos de una oficina sean insertados correctamente. A continuación se muestra en la figura 13 el resultado de la ejecución de esta prueba:



*Figura 13 Resultado de la prueba unitaria realizada a la funcionalidad CrearOficina()*

### 3.3.2 Pruebas de Caja Negra (funcional)

Así como las pruebas de caja blanca son aplicadas a los requisitos funcionales desde el interior del módulo, las de caja negra están dirigidas a las funciones externas. Una vez conocida la función específica para la que fue diseñada el producto, pretenden demostrar que las funciones del software son totalmente operativas y al mismo tiempo busca errores en cada función. Estas pruebas examinan algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta su estructura lógica interna.

#### **Diseño de casos de pruebas:**

Un caso de pruebas es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de prueba especifican una forma de probar el sistema, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que ha de probarse. Se realizan con el fin de asegurar que el producto es operativo.

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

Para comprobar el correcto funcionamiento del módulo de administración se definieron un conjunto de casos de pruebas que reúnen la mayor cantidad de escenarios posibles. A continuación se muestra el caso de prueba Gestionar Rol, los restantes se encuentran en el ANEXO 6: "Casos de Prueba de Caja Negra".

### Caso de prueba Gestionar Rol:

Descripción general: Permite al usuario adicionar, modificar y mostrar los datos de las personas que posean dicho rol

Condiciones de ejecución: El usuario debe estar autenticado con el rol de administrador.

### Tablas de los escenarios:

#### SC1: Añadir Rol.

Escenario	Descripción	Nombre	Descripción	Estado	Respuesta del sistema	Flujo central
<i>EC 1.1 Se añade el rol satisfactoriamente.</i>	<i>El usuario introduce los datos del rol a registrar.</i>	V	V	V	<i>El sistema inserta el rol con los datos especificados.</i>	<i>1-El usuario selecciona la opción "Gestionar Rol". 2-El sistema muestra la interfaz "Gestionar Rol". 3-Selecciona la opción "Añadir Rol" 4-Entra los datos solicitados. 5-Da clic en el botón "Salvar".</i>

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA.:.

<p><i>EC 1.2 Campos Vacíos.</i></p>	<p><i>El usuario deja de insertar algún dato.</i></p>	<p>V</p>	<p>NA</p>	<p>V</p>	<p><i>Se muestra un mensaje al lado del campo sin llenar con el siguiente texto: "Debe introducir los datos requeridos".</i></p>	<p>1-El usuario selecciona la opción "Gestionar Rol". 2-El sistema muestra la interfaz de "Gestionar Rol". 3- Selecciona la opción "Añadir Rol" 4-Entra los datos solicitados dejando alguno vacío. 5-Da clic en el botón "Salvar".</p>
<p><i>EC 1.3 Caracteres Extraños</i></p>	<p><i>El usuario introduce datos con caracteres extraños.</i></p>	<p>I</p>	<p>V</p>	<p>V</p>	<p><i>Se muestra un mensaje al lado del campo incorrecto: "Debe introducir los datos correctamente".</i></p>	<p>1-El usuario selecciona la opción "Gestionar Rol". 2-El sistema muestra la interfaz de "Gestionar Rol". 3- Selecciona la opción "Añadir Rol" 4-Entra los datos solicitados de forma errónea. 5-Da clic en el botón "Salvar".</p>

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

*Tabla 6 Escenario "Añadir Rol" del caso de prueba "Gestionar Rol"*

### SC2: Modificar Rol.

El escenario Modificar Rol funciona de manera similar al escenario Crear Rol en caso de introducir caracteres incorrectos o dejar campos vacíos debido a que posee los mismos campos.

Escenario	Descripción	Nombre	Descripción	Estado	Respuesta del sistema	Flujo central
<i>EC 2.1 Se modifica el rol satisfactoriamente.</i>	<i>El usuario modifica los datos del rol seleccionado.</i>	V	V	V	<i>El sistema actualiza los datos en la base de datos y en el listado de roles.</i>	<i>1-El usuario selecciona la opción "Gestionar Rol".</i> <i>2-El sistema muestra la interfaz de "Gestionar Rol".</i> <i>3-Selecciona la opción "Modificar Rol"</i> <i>4-Entra los datos solicitados.</i> <i>5-Da clic en el botón "Salvar".</i>

*Tabla 7 Escenario "Modificar Rol" del caso de prueba "Gestionar Rol"*

### SC3: Mostrar Información

Escenario	Descripción	Nombre	Descripción	Estado	Respuesta del sistema	Flujo central
<i>EC 3.1 Se muestran los datos de las personas que posean un rol seleccionado.</i>	<i>El usuario selecciona esta opción para mostrar más información sobre un rol.</i>	V	V	V	<i>El sistema despliega la información solicitada del rol seleccionado.</i>	<i>1-El usuario selecciona la opción "Gestionar Rol".</i> <i>2-El sistema muestra la interfaz de</i>

## .:CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA:.

						<p>“Gestionar Rol”.</p> <p>3-Selecciona un rol de la lista de roles.</p> <p>4-Selecciona la opción “Mostrar Información”</p>
--	--	--	--	--	--	--

**Tabla 8 Escenario “Mostrar Información” del caso de prueba “Gestionar Rol”**

### Descripción de las variables:

A continuación se describen las variables implicadas en el Caso de Prueba Gestionar Rol para un mayor conocimiento de lo que el sistema espera en cada campo y/o opción.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	En este campo se insertará el nombre del rol. No se deben introducir números, ni caracteres especiales.
2	Descripción	Text área	No	En este campo se insertarán la descripción del rol. No se deben introducir números, ni caracteres especiales.
3	Estado	Selección simple	Si	Esta opción estará activo por defecto, pudiera desactivarse o no haciendo clic sobre él.

**Tabla 9 Descripción de las variables del caso de prueba “Gestionar Rol”**

### No conformidades detectadas:

Se realizaron alrededor de 4 iteraciones, donde se identificaron algunas no conformidades (ver figura 14), las cuales fueron disminuyendo considerablemente con cada iteración realizada:



*Figura 14 Iteraciones de la prueba de caja negra*

### 3.3.3 Resultado General de las Pruebas

Los resultados arrojados por el sistema al final de cada prueba aplicada han sido satisfactorios (teniendo en cuenta que el descubrimiento de un defecto significa un éxito para la mejora de la calidad), indicando el nivel de calidad y eficiencia que posee el mismo. La aplicación de las pruebas aprovechó para verificar las funcionalidades más importantes de módulo de administración.

### 3.4 Conclusiones Parciales

En este capítulo se han desarrollado los aspectos que muestran la implementación del sistema. Con el modelado del diagrama de componentes quedó representada la estructura lógica del sistema. Se modeló el diagrama de despliegue, en el cual se visualiza cómo estará distribuido el sistema en un plano físico. Se realizaron pruebas para comprobar el correcto funcionamiento del sistema validando que el mismo cumple los requisitos establecidos y garantizando de esta forma su calidad.



## CONCLUSIONES

Como resultado de la investigación realizada se arriba a las siguientes conclusiones:

- El análisis a los sistemas de identificación realizado fundamentó la necesidad del desarrollo del sistema.
- La utilización de la metodología seleccionada posibilitó documentar la solución, lo que brindó guías en el proceso de desarrollo y agilizó la implementación del mismo.
- El modelo de dominio definido permitió tener un amplio conocimiento del funcionamiento del negocio.
- Con la definición de los requisitos del sistema se especificaron las características que el mismo debe cumplir.
- La implementación del módulo de administración permitió el cumplimiento de los requisitos definidos para la solución del software.
- Durante el desarrollo del módulo de administración se realizaron pruebas de caja blanca y de caja negra al mismo, las cuales permitieron comprobar su correcto funcionamiento.
- La solución obtenida permite que en la Plataforma Modular de Identificación y Control de Acceso, sean gestionados los elementos de configuración y seguridad.

## RECOMENDACIONES

Luego de haber desarrollado una primera versión del módulo de administración se recomienda hacer uso de la propuesta implementada, que con su integración a la PMICA, se logrará la gestión de los elementos de configuración y seguridad.

## BIBLIOGRAFÍA REFERENCIADA

1. Definición ABC. [En línea] [Citado el: 13 de 02 de 2013.] <http://www.definicionabc.com/general/identificacion.php>.
2. SlideShare. [En línea] [Citado el: 13 de 02 de 2013.] <http://www.slideshare.net/JohanPrevotR/gestion-de-la-configuracion-del-software-10471407>.
3. **Achillo, Edgar Pedro Garcia**. Viceministerio Ciencia y Tecnología. [En línea] [Citado el: 13 de 02 de 2013.] <http://www.cienciaytecnologia.gob.bo/vcyt2012/uploads/cap-2-funciones-y-procedimientos.pdf>.
4. Promonegocios.net. [En línea] [Citado el: 13 de 02 de 2013.] <http://www.promonegocios.net/administracion/definicion-administracion.html>.
5. **Pérez, José Alfredo Hernández**. *Módulo de Administración en un Sistema Informático*.
6. **Reina Consuegra, Susana y Legón Campo, Adonis César**. *Trabajo de Diploma: Módulo de Administración para el SAIME de la República Bolivariana de Venezuela*. La Habana : s.n., 2007.
7. **Venero Paez, Diana y Valdés González, Alejandro**. *Trabajo de diploma: Módulo de Administración para el Sistema Único de Identificación Nacional de la República de Cuba*. La Habana : s.n., 2010.
8. **Molpeceres, Alberto**. *Procesos de Desarrollo: Rup, XP y FDD*. 2002.
9. Scribd. [En línea] [Citado el: 16 de 02 de 2013.] <http://es.scribd.com/doc/2080534/UML>.
10. ENI. [En línea] [Citado el: 16 de 02 de 2013.] <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.
11. [En línea] [Citado el: 16 de 02 de 2013.] <http://www.utm.mx/~caff/doc/CASEparaGenerarCodigoCconDiagramasUML.pdf>.
12. **Nobrega, María de**. [En línea] [Citado el: 16 de 02 de 2013.] [http://curso\\_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php](http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php).
13. **Pozos Mendez, Pablo Esteban**. Segundo trabajo: Herramienta Case Visual Paradigm | Herramientas Automatizadas. [En línea] 17 de Febrero de 2012. [Citado el: 24 de Enero de 2013.] <http://dianbeel.blogspot.com/2012/06/segundo-trabajo-herramienta-case-visual.html>.
14. Carlos Blanco. [En línea] [Citado el: 03 de 02 de 2013.] <http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/>.
15. Visual Studio. [En línea] [Citado el: 06 de 01 de 2013.] <http://msdn.microsoft.com/es-es/library/vstudio/fx6bk1f4%28v=vs.100%29.aspx>.

## .:BIBLIOGRAFÍA REFERENCIADA:.

16. MSDN. [En línea] [Citado el: 03 de 01 de 2013.] <http://msdn.microsoft.com/es-es/library/h8w79z10%28v=vs.80%29.aspx>.
17. **Echeverry, Natalia y Pérez Restrepo, Jhon Fredy.** MSDN Microsoft. [En línea] [Citado el: 13 de Enero de 2013.] <http://msdn.microsoft.com/es-es/library/vstudio/z1zx9t92%28v=vs.100%29.aspx>.
18. **Manuel Resinas de Reyna, Manuel J. Recena Soto.** *Introducción a los Sistemas de Control de Versiones.* Universidad de Sevilla. 2005.
19. TortoiseSVN. [En línea] [Citado el: 20 de Febrero de 2013.] [tortoisesvn.net/docs/nightly/TortoiseSVN\\_es/help-onepage.html](http://tortoisesvn.net/docs/nightly/TortoiseSVN_es/help-onepage.html).
20. **Cuevas Guerrero, Daniel.** *La Plataforma Microsoft .NET.*
21. MSDN Microsoft. [En línea] [Citado el: 20 de Febrero de 2013.] <http://msdn.microsoft.com/es-es/library/w0x726c2%28v=vs.100%29.aspx>.
22. w3schools. [En línea] [Citado el: 17 de Enero de 2013.] <http://www.w3schools.com/aspnet/default.asp>.
23. Programación en castellano. [En línea] [Citado el: 24 de Febrero de 2013.] [http://www.programacion.com/articulo/conceptos\\_basicos\\_de\\_orm\\_object\\_relational\\_mapping\\_349](http://www.programacion.com/articulo/conceptos_basicos_de_orm_object_relational_mapping_349).
24. **Nicolas Schenker, Gabriel y Cure, Aaron.** *NHibernate 3 Beginner's Guide.* 2011.
25. PostgreSQL-es. [En línea] [Citado el: 20 de Enero de 2013.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
26. Tecnología y Synergix. [En línea] [Citado el: 18 de Febrero de 2013.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
27. Office. [En línea] [Citado el: 20 de 03 de 2013.] <http://office.microsoft.com/es-es/access-help/conceptos-basicos-del-diseno-de-una-base-de-datos-HA001224247.aspx>.
28. **Visconti, M. y Astudillo, H.** *Fundamentos de Ingeniería de Software.* [En línea] [Citado el: 10 de Marzo de 2013.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
29. **Gamma, Erich, y otros, y otros.** *Patrones de Diseño.*

## BIBLIOGRAFÍA CONSULTADA

1. Repositorio Institucional. [En línea] 2013. [http://repositorio\\_institucional.uci.cu/jspui/](http://repositorio_institucional.uci.cu/jspui/).
2. Bases de Datos Bibliográficas. [En línea] 2013. <http://biblioteca2.uci.cu/?q=base-datos>.
3. Gemalto. [En línea] <http://www.gemalto.com/>.
4. Sagem. [En línea] <http://www.sagem.com>.
5. Data Card. [En línea] <http://www.datacard.com/>.
6. **Reina Consuegra, Susana y Legón Campo, Adonis César.** *Módulo de Administración para el Sistema SAIME de la República Bolivariana de Venezuela.* 2007.
7. **Venero Paez, Diana y Valdés González, Alejandro.** *Módulo de Administración para el Sistema Único de Identificación Nacional de la República de Cuba.* 2010.
8. **Díaz Fernández, Alain.** *Propuesta de informatización del proceso de identificación de personas en Cuba.* 2009.
9. *Identificación-Autenticación-Autorización.* s.l. : EVA.
10. *ID SyStemS. Innovative solutions to protect your identity.*
11. **Jean Paul V.A.** *Design Patterns in C#.* 2012.

### GLOSARIO DE TÉRMINOS

#### -A-

**ASP.NET:** Marco de desarrollo *web* unificado para la creación de páginas y sitios *web* con HTML, CSS, JavaScript y secuencias de comandos del servidor.

**ADO.NET:** Es un conjunto de componentes y clases bases que permiten a los programadores la interacción con base de datos y otras fuentes.

#### -B-

**BSD:** Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva. Posee menos restricciones en comparación con otras como la GPL. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

#### -C-

**CISED:** Centro de Identificación y Seguridad Digital.

**Credencial:** Documento de identificación que acredita como personal autorizado dentro de la UCI.

**CASE (Computer Automated System Engineering):** Conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo

de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

**C#:** Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

**CVC:** Es el sistema de control de versiones concurrentes más utilizado en el mundo del software libre, sin embargo es bastante y tiene bastantes carencias, como son la falta de versionado de directorios, de renombrado o de copias; su uso poco eficiente de la red o el que no almacene archivos binarios completos, solo los cambios entre versiones.

**CSS (Cascading Style Sheets):** Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

**CLR:** Es el entorno de tiempo en ejecución que proporciona .NET Framework que ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo.

#### -E-

**Excepciones:** Errores que ocurren en procesos internos de la aplicación que deben ser tratados para no caer en estados inestables y brindar seguridad y confianza al usuario.

#### -F-

## .:GLOSARIO DE TÉRMINOS.:.

### **FDD (*Feature Driven Development*):**

Metodología de software utilizada en el proceso de desarrollo de la PMICA.

**Framework:** Conjunto de clases modeladas de forma general para resolver problemas relacionados en un contexto específico.

**Fuente de datos:** Toda materia que genere, contenga, suministre o transfiera datos. Lugar de donde se obtendrán los datos a utilizar en el sistema. Puede ser una base de datos, fichero, servicio *web*, etc.

### -H-

### **HTML (*HyperText Markup Language*):**

Lenguaje compuesto de una serie de etiquetas o marcas que permiten definir el contenido y la apariencia de las páginas *web*.

### -I-

**Identificación:** Acción por parte de un usuario de presentar su identidad a un sistema, usualmente se usa un identificador de usuario.

**Identación:** Mover un bloque de texto hacia la derecha insertando espacios o tabuladores para separarlo del texto adyacente.

### -L-

**LDAP (*Lightweight Directory Access Protocol*):** Protocolo Ligero de Acceso a Directorios que permite administrar directorios, esto es, acceder a bases de información de usuarios de una red.

### -M-

**Módulo:** Una parte de un programa que lleva a cabo una función específica y puede ser utilizado solo o combinado con otros módulos del mismo programa.

### -R-

**Repositorio:** Sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

### -P-

**Plataforma:** Sistema que sirve como base para hacer funcionar determinados módulos de *hardware* o de *software* con los que es compatible. Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles.

**PMICA:** Plataforma Modular de Identificación y Control de Acceso.

**PVC:** El PVC (policloruro de vinilo) es una combinación química de carbono, hidrógeno y cloro. Es el plástico con menos dependencia del petróleo. Las impresoras de credenciales pvc son ideales para la credencialización en escuelas, centros de gobierno, control de seguridad, control de asistencia, clubs de membresías y eventos deportivos.

## ANEXOS

## ANEXO 1. Interfaces de los sistemas estudiados.

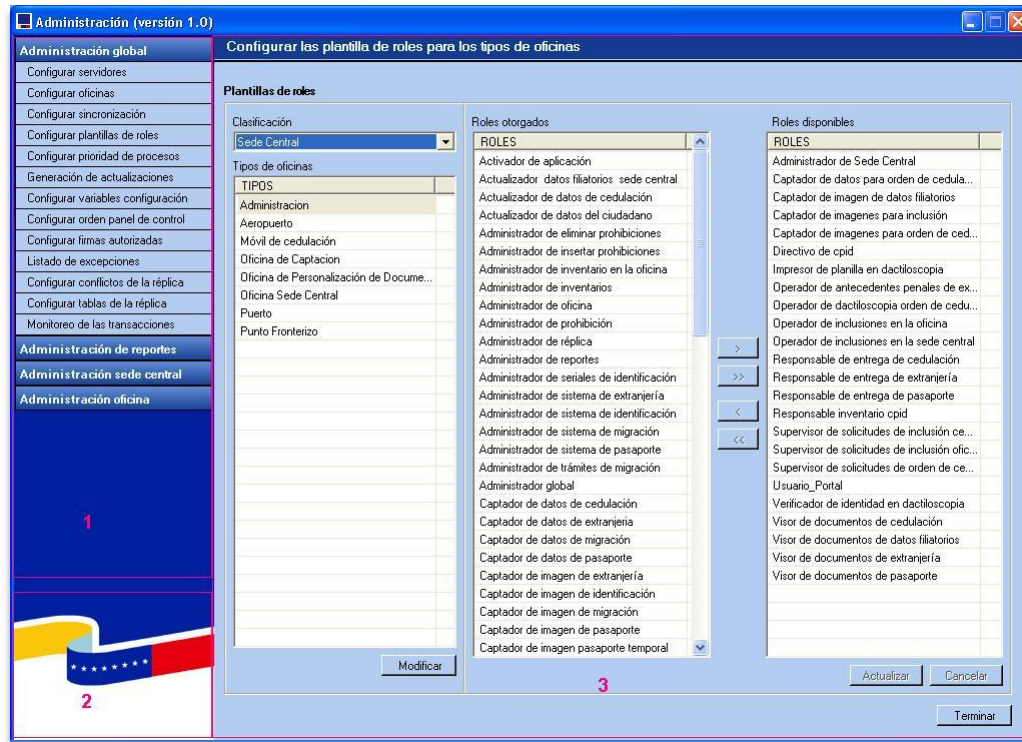


Figura 15 Interfaz principal del sistema SAIME



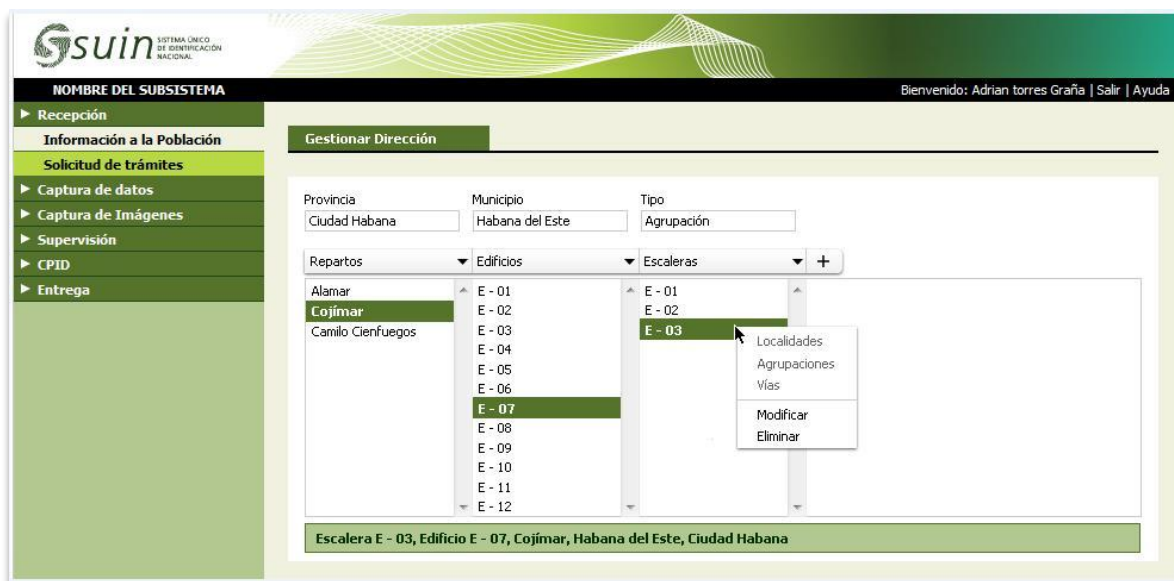
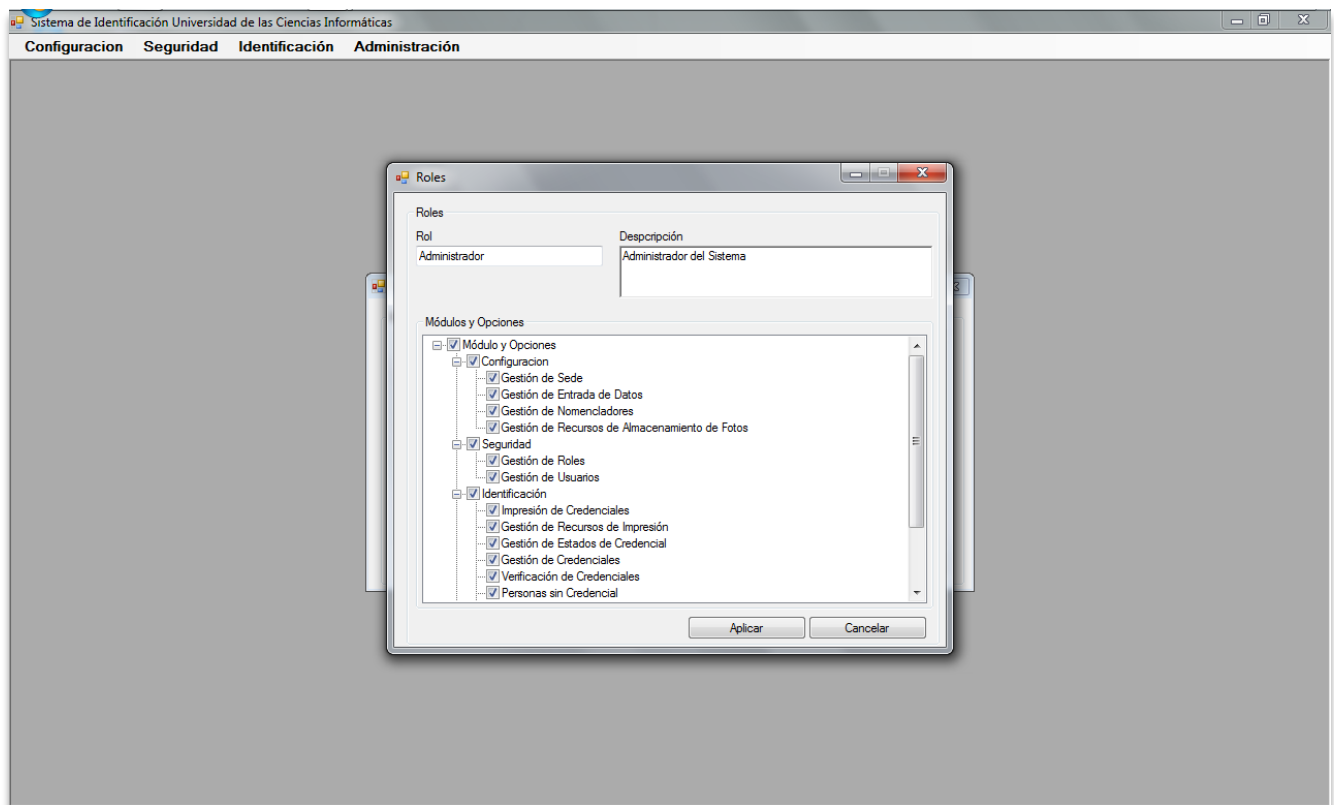


Figura 16 Interfaz "Crear Dirección" del sistema SUIN



Figura 17 Interfaz "Autenticar" del Sistema de Identificación de la UCI



**Figura 18** Interfaz "Gestionar Roles" del Sistema de Identificación de la UCI

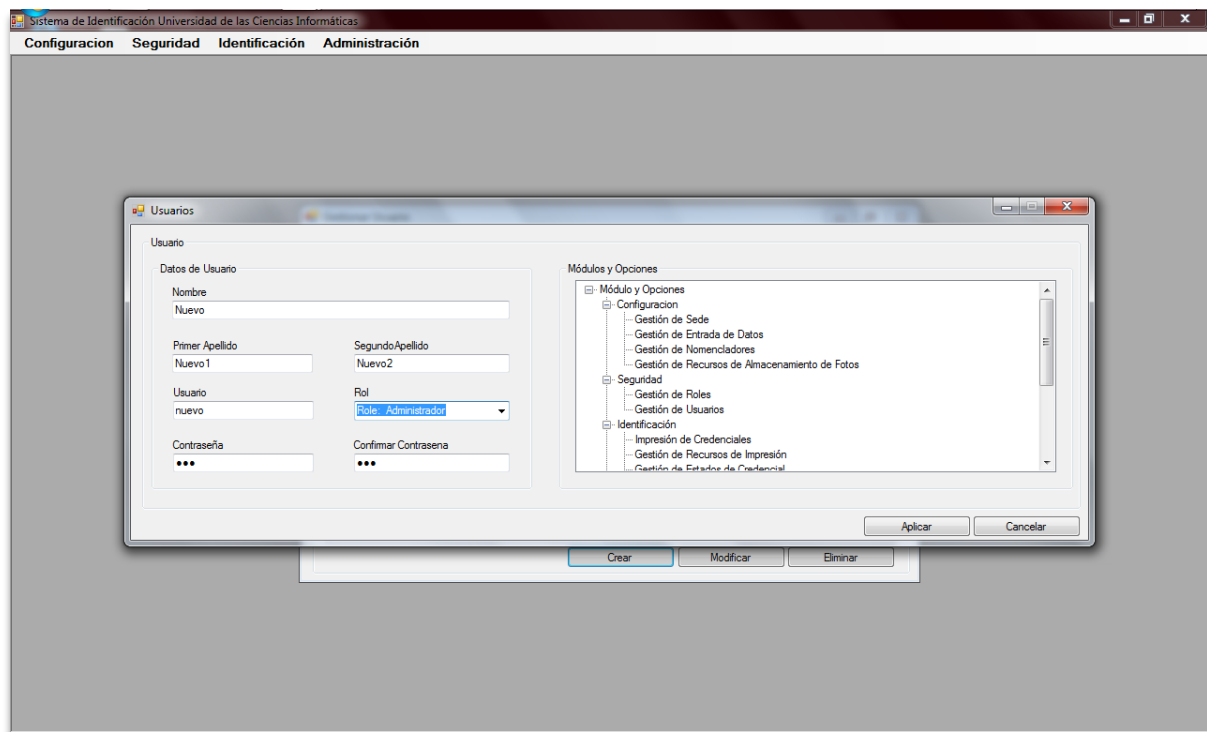


Figura 19 Interfaz "Gestionar Usuarios" del Sistema de Identificación de la UCI

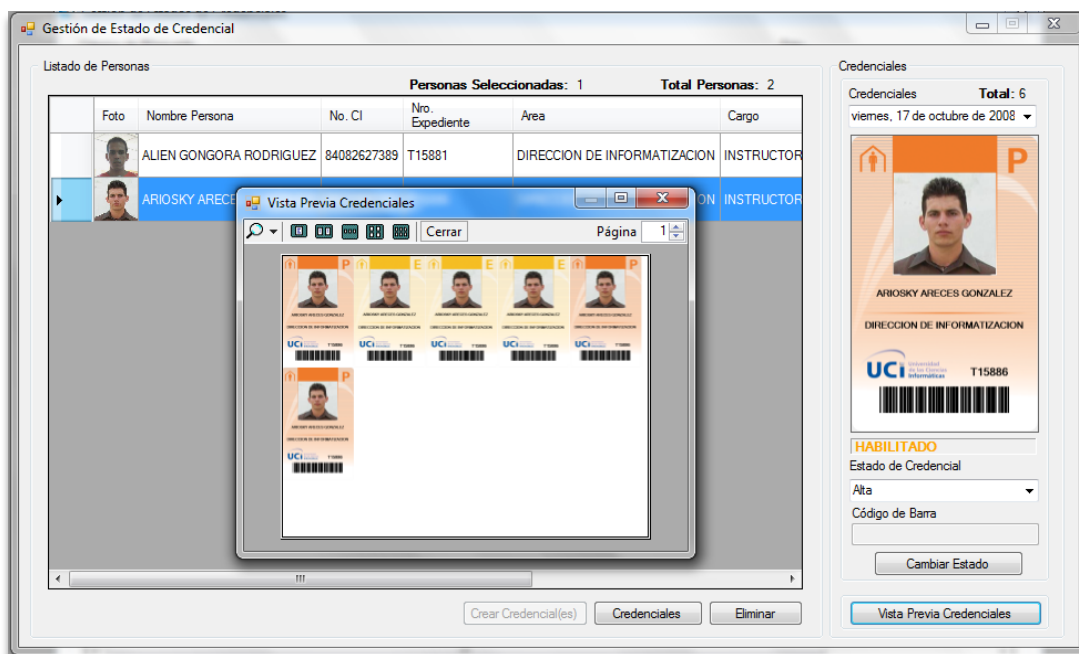


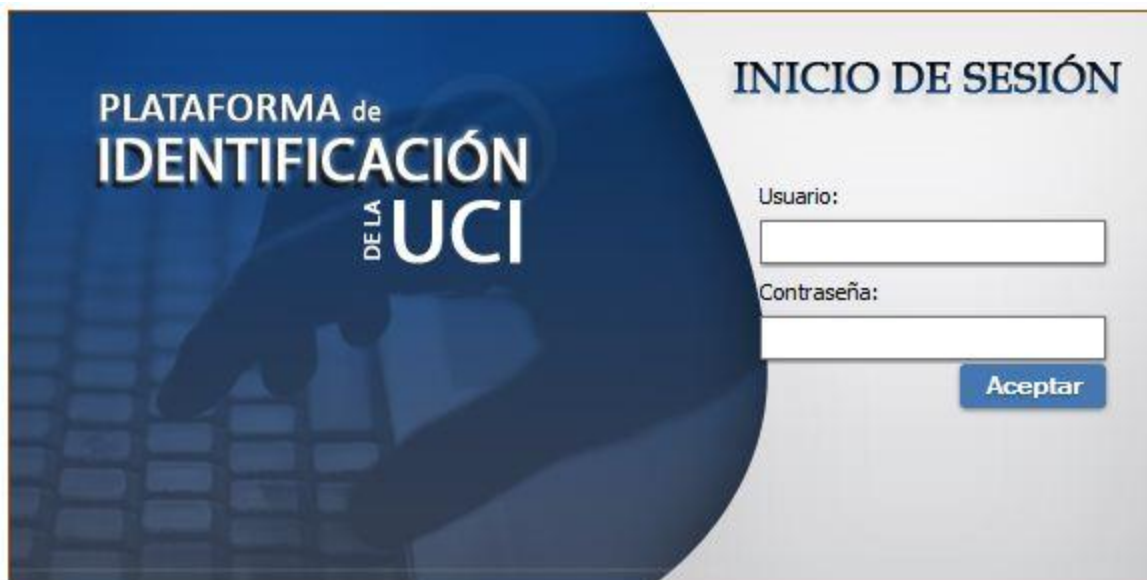
Figura 20 Interfaz "Gestión de Estado de Credencial" del Sistema de Identificación de la UCI

## ANEXO 2. Descripción de las funcionalidades.

Autenticar Usuario:

Precondiciones	
Funcionalidades tratadas	RF1, RF1.1, RF1.2, RF1,3
Conceptos tratados	Usuario, Contraseña
Descripción básica	<p>1 El sistema muestra una interfaz de inicio de sección del sistema (ver figura 1) donde el usuario solicitará la entrada al sistema introduciendo su nombre de usuario y contraseña.</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>-Usuario</li> <li>-Contraseña</li> <li>-Dominio</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Aceptar</li> </ul> <p>1.2 Luego del usuario introducir los datos requeridos da clic en el botón "Aceptar" y el sistema verifica que la corrección de los mismos.</p> <p>1.2.1 Si los datos son correctos se configura el sistema según el rol del usuario, se define el nivel de acceso que este tendrá, y se le permite entrar al sistema mostrando un mensaje de bienvenida.</p> <p>1.2.2 Si los datos son incorrectos se muestra un mensaje: "Usuario o contraseña incorrecto" y se le denegará la entrada al sistema. (ver figura 2)</p> <p>1.3 El usuario selecciona el dominio por el cual desea entrar al sistema. Seleccionará el dominio LDAP de la UCI si se está autenticado por primera vez y en caso de ya pertenecer al sistema se autenticará por el dominio local.</p> <p>Si el usuario trata de entrar por el dominio local sin pertenecer al mismo se muestra un mensaje "Pruebe con el dominio UCI". De igual manera sucederá en caso de intentar entrar por el dominio UCI perteneciendo ya el dominio local (ver figura 3).</p>
Prototipos	

Figura 1. Autenticar:



PLATAFORMA de IDENTIFICACIÓN DE LA UCI


## INICIO DE SESIÓN

Usuario:

Contraseña:

**Aceptar**

Figura 2. Usuario y contraseña incorrecto:



PLATAFORMA de IDENTIFICACIÓN DE LA UCI

## INICIO DE SESIÓN

Usuario:

Contraseña:

Dominio UCI  Local

**Aceptar**

**Usuario o Contraseña Incorrecto.**

Figura 3. Dominio incorrecto:

Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	El usuario podrá acceder o no al sistema.

*Tabla 10 Descripción de la funcionalidad "Autenticar Usuario"*

Gestionar Ciudadanos:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador.
Funcionalidades tratadas	RF2: RF2.1, RF2.2, RF2.3
Conceptos tratados	Ciudadano, Sistema
Descripción básica	<p>1 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción "Gestionar Ciudadano" el sistema muestra la interfaz "Gestionar Ciudadano". (Ver figura 2)</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>- Ci</li> <li>-Solapín</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>- Buscar</li> <li>- Crear</li> </ul> <p>2.1 Si el usuario introduce el ci del ciudadano que desea buscar en la base de datos y da clic en el botón "Buscar" el sistema muestra los datos del ciudadano. (ver figura 2) Igualmente si introduce el número de solapín de ciudadano.</p> <p>2.2 Si se da clic en el botón "Crear" se adiciona el ciudadano a la base de datos del sistema. (ver figura 3)</p>



Figura 1. Gestionar Ciudadano:

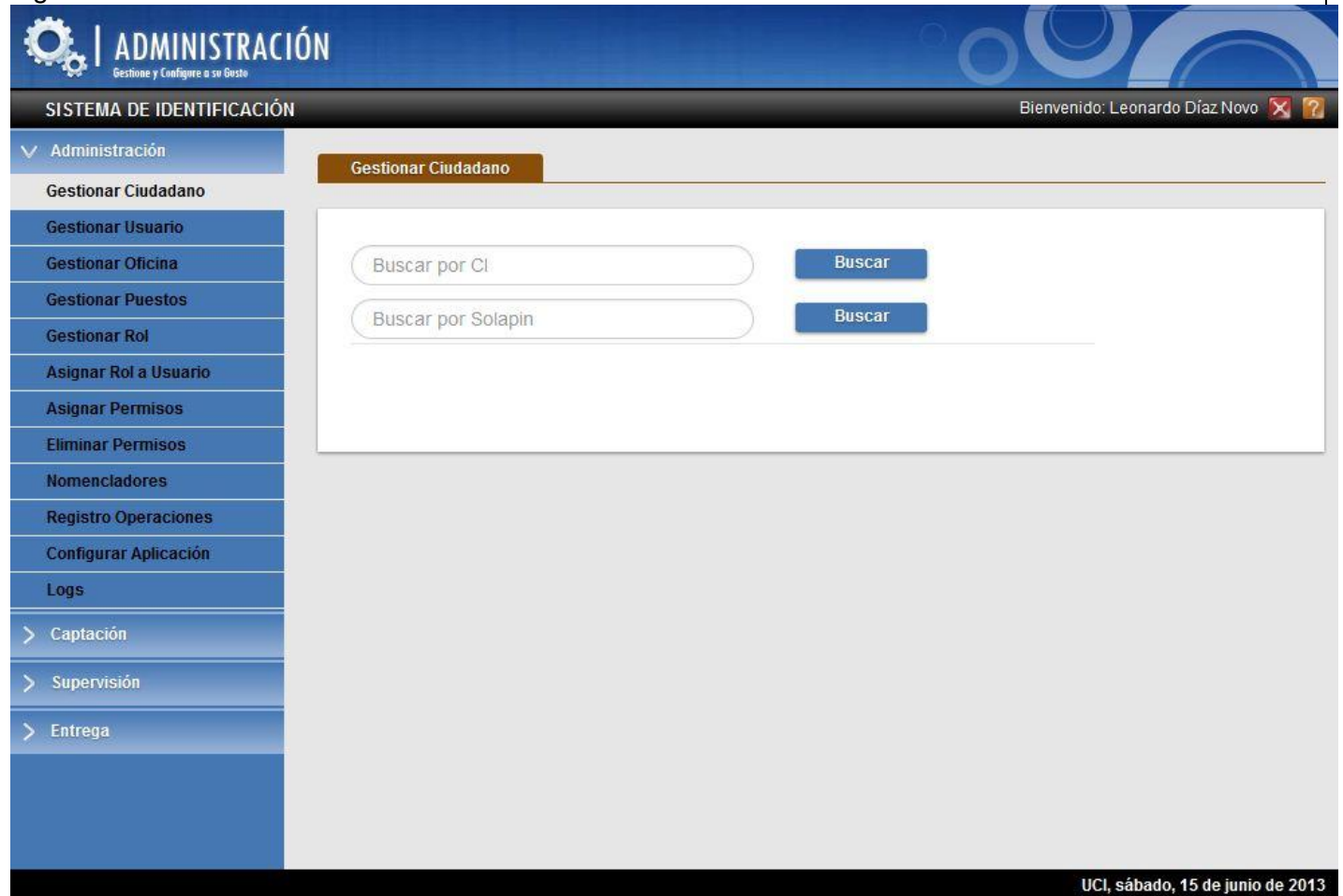


Figura 2. Crear Ciudadano:



**ADMINISTRACIÓN**  
Gestione y Configure a su Gusto

SISTEMA DE IDENTIFICACIÓN Bienvenido: Leonardo Díaz Novo

Administración

Gestionar Ciudadano

Gestionar Usuario

Gestionar Oficina

Gestionar Puestos

Gestionar Rol

Asignar Rol a Usuario

Asignar Permisos

Eliminar Permisos

Nomencladores

Registro Operaciones

Configurar Aplicación

Logs

> Captación

> Supervisión

> Entrega

Gestionar Ciudadano

91010420978 Buscar

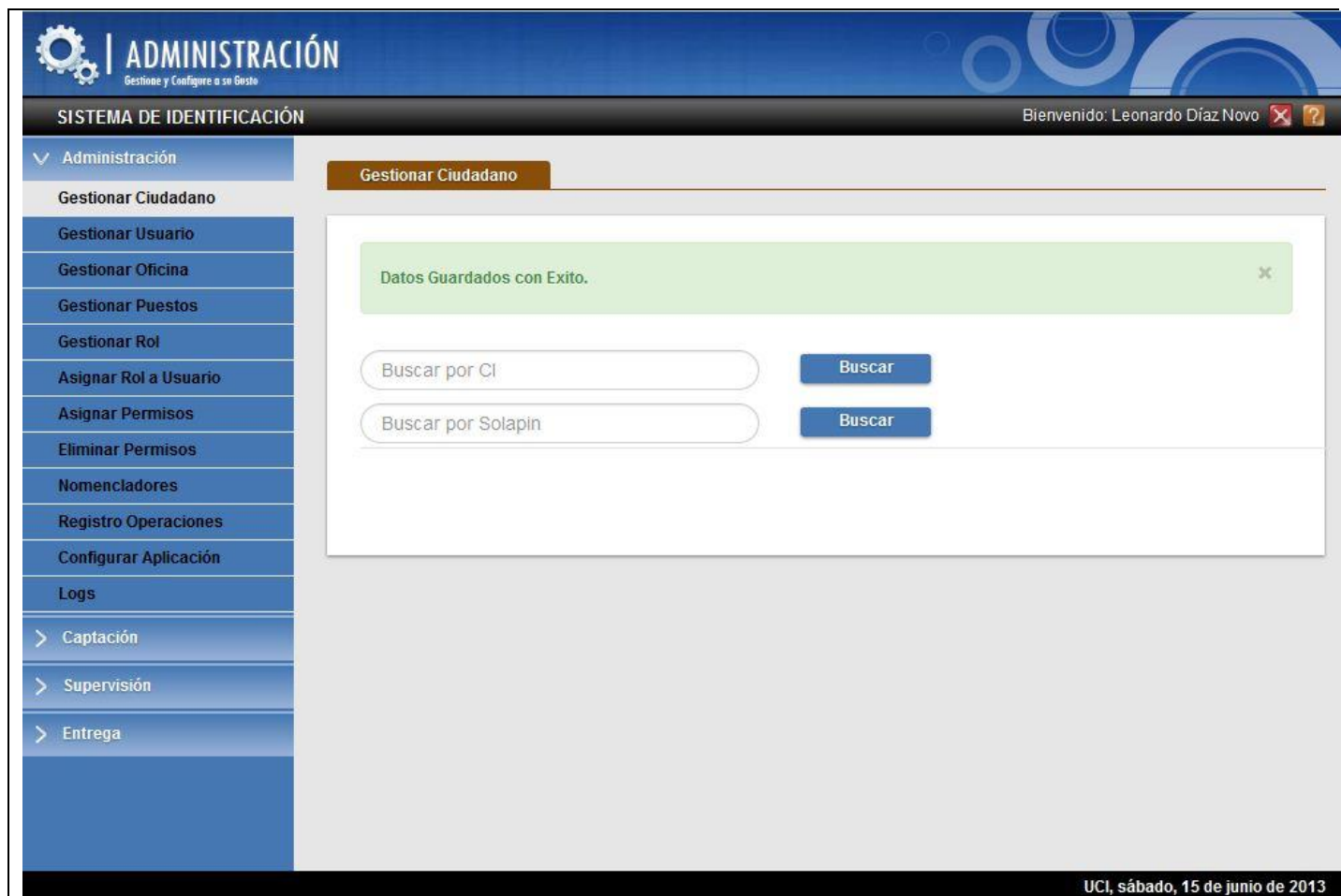
Buscar por Solapín Buscar

Foto	Nombre	S_Nombre	Apellido	S_Apellido	CI	Sexo	Categoría	F_Nac
	Dailen		Barrera	Millet	91010420978	F	Estudiante	04/01/1991

Crear

UCI, sábado, 15 de junio de 2013

Figura 3. Ciudadano creado con éxito:



Descripción alterna:	
Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Se añadirán los ciudadanos al sistema.

**Tabla 11 Descripción de la funcionalidad "Gestionar Ciudadano"**

## Gestionar Usuario:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador. Los usuarios deben estar previamente creados.
Funcionalidades tratadas	RF3, RF3.1, RF3.2, RF3.3
Conceptos tratados	Usuario, Ciudadano
Descripción básica	<p>1 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción "Gestionar Usuario" el sistema muestra la interfaz "Gestionar Usuario" (ver figura 1) donde se visualizará un listado con los usuarios pertenecientes al sistema y sus respectivos datos.</p> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Editar</li> <li>-Mostrar Información</li> </ul> <p>2.1 Si el usuario selecciona la opción "Editar" se mostrará la interfaz "Editar Usuario" (ver figura 2) donde podrá cambiar el estado de dicho usuario.</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>-Estado</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Salvar</li> <li>-Cancelar</li> </ul> <p>2.1.2 Si se da clic en el botón "Salvar" se modificará el estado del usuario seleccionado.</p> <p>2.1.3 Si se da clic en el botón "Cancelar" se cerrará la ventana "Editar Usuario".</p> <p>2.2 Si el usuario selecciona la opción "Mostrar información" se desplegará del usuario correspondiente los datos correspondientes al ciudadano al que pertenece dicho usuario. (Ver figura 3)</p>
Prototipos	

Figura 1. Gestionar Usuario:

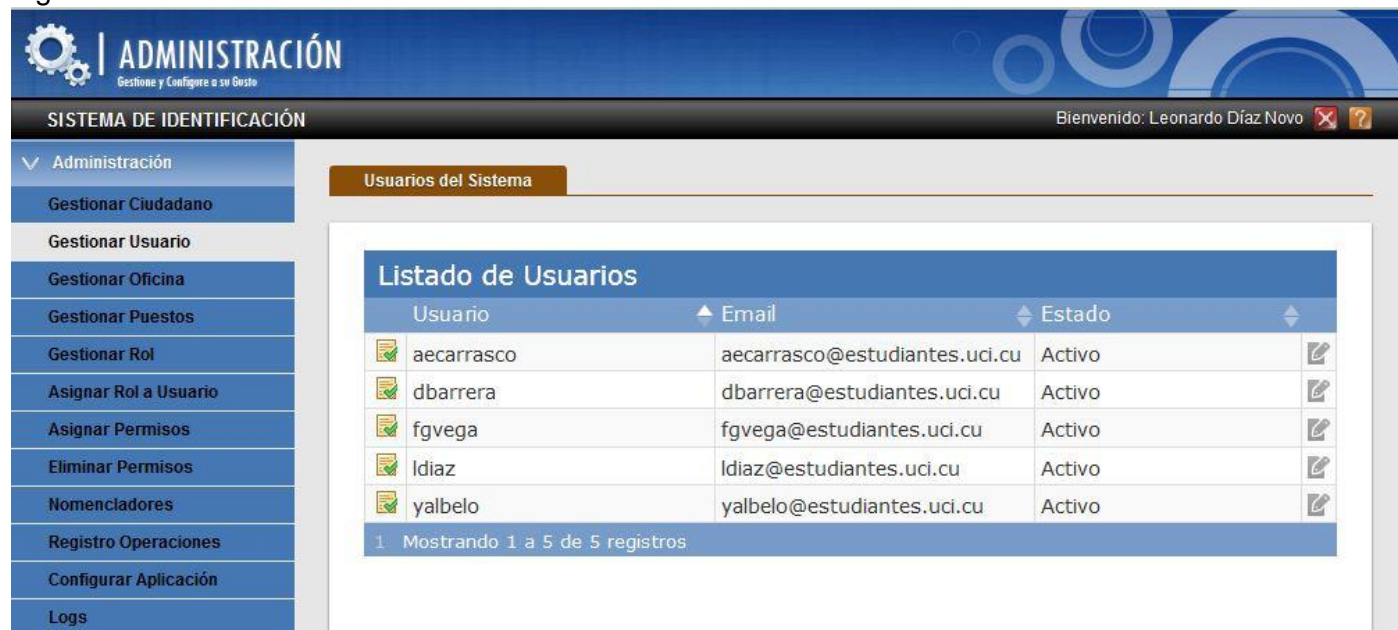


Figura 2. Editar Usuario:

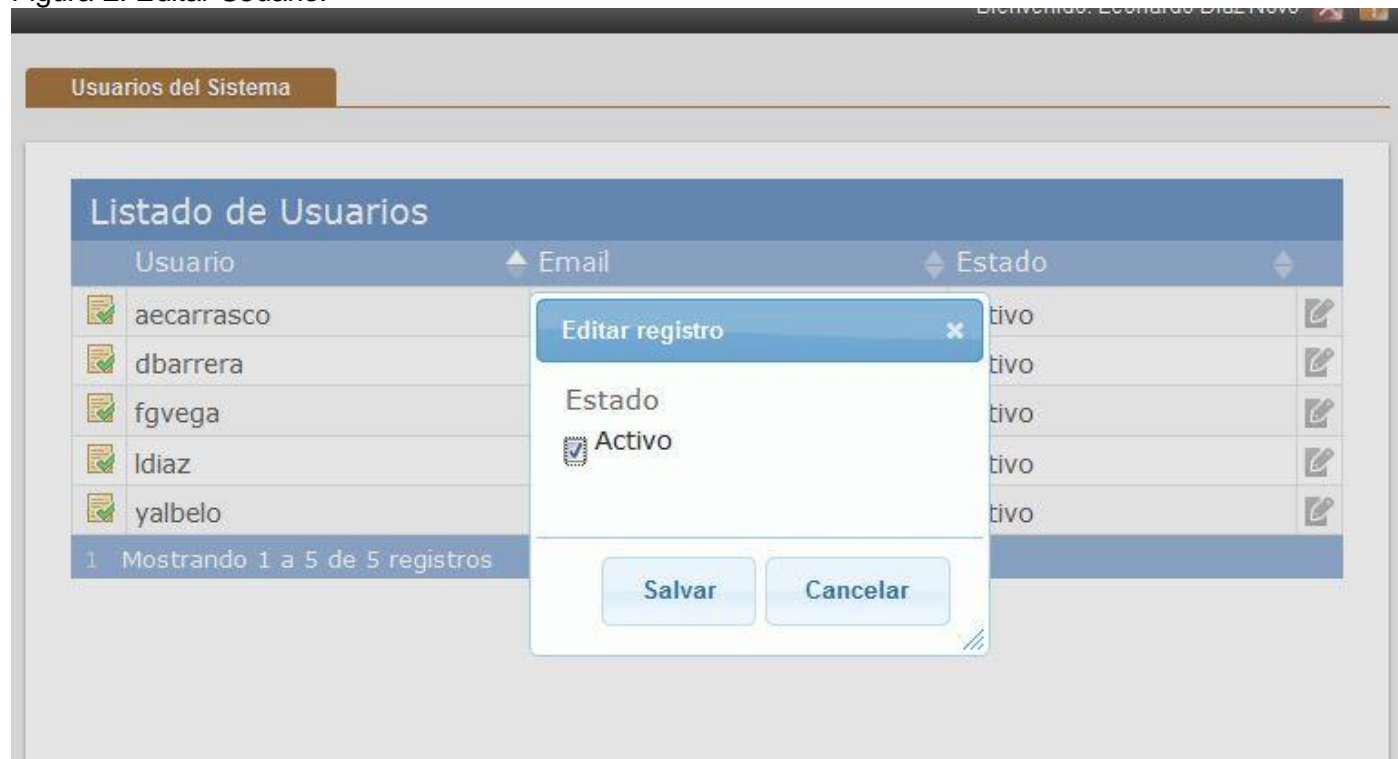



Figura 3. Mostrar Información:





Usuarios del Sistema

### Listado de Usuarios

Usuario	Email	Estado
 aecarrasco	aecarrasco@estudiantes.uci.cu	Activo

#### Datos del Usuario - aecarrasco

Nombre de la Persona	Categoría	Ci	Sexo
Asiel Echemendia Carrasco	estudiante	89053034184	Masculino

 dbarrera	dbarrera@estudiantes.uci.cu	Activo
 fgvega	fgvega@estudiantes.uci.cu	Activo
 ldiaz	ldiaz@estudiantes.uci.cu	Activo
 yalbelo	yalbelo@estudiantes.uci.cu	Activo

1 Mostrando 1 a 5 de 5 registros

## Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Se mostrarán los datos de los usuarios y se podrán editar el estado de los mismos.

**Tabla 12 Descripción de la funcionalidad "Gestionar Usuario"**

## Asignar Permiso a Rol:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador. Los roles y usuarios deben estar previamente creados.
Funcionalidades tratadas	RF5, RF5.1, RF5.2, RF5.3
Conceptos tratados	Rol, Usuario, Módulo
Descripción básica	<p>2 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> </ul>

	<ul style="list-style-type: none"> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción "Asignar Permiso a Rol" el sistema muestra la interfaz "Asignar Permiso a Rol" (ver figura 1) con el siguiente campo a llenar:  <b>Campos a llenar:</b>          -Usuario  <b>Acciones:</b>          -Ver</p> <p>2.1 Si el usuario introduce el dato requerido y selecciona la opción "Ver", se mostrarán los roles pertenecientes a dicho usuario y los módulos del sistema.  <b>Acciones:</b>          -Mostrar          -Guardar</p> <p>2.1.2 Al usuario seleccionar un módulo se mostrarán las partes de ese módulo (ver figura 2) a las que se le podrán dar permisos al usuario seleccionado. El usuario marca las opciones que desea y da clic en "Guardar" para salvar las operaciones realizadas.</p>
--	--

## Prototipos

Figura 1. Asignar Permiso a Rol:

The screenshot displays a web interface titled "Activación de Módulos". On the left, under the heading "Módulos", there is a list of four items, each with a checked checkbox: "Administración", "Captación", "Supervisión", and "Entrega". To the right of this list, there is a section labeled "Seleccione el Tema" containing a dropdown menu with "Clásico" selected. Below the dropdown menu is a blue "Guardar" button. At the bottom left of the main content area, there is another blue "Guardar" button.

Figura 2. Asignar Permiso a Rol (módulos):



Asignar Permisos.

Rol:  
 Módulo:

Captación de Datos  
 Captación de Imágenes

Descripción alterna:	
Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Quedarán asignados los permisos a los roles seleccionados.

**Tabla 13 Descripción de la funcionalidad "Asignar Permiso a Rol"**

#### Eliminar Permiso:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador. Los roles y usuarios deben estar previamente creados.
Funcionalidades tratadas	RF5.3
Conceptos tratados	Rol, Usuario, Módulo
Descripción básica	<p>3 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> </ul>

	<ul style="list-style-type: none"> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción "Eliminar Permiso" el sistema muestra la interfaz "Eliminar Permiso" (ver figura 1) con el siguiente campo a llenar:</p> <p><b>Campos a llenar:</b> -Usuario</p> <p><b>Acciones:</b> -Ver</p> <p>2.1 Si el usuario introduce el dato requerido y selecciona la opción "Ver", se mostrarán los permisos pertenecientes a dicho usuario.</p> <p><b>Acciones:</b> -Seleccionar -Guardar</p> <p>2.1.2 Al usuario seleccionar el permiso que desea retirarle al usuario buscado y da clic en "Guardar" para salvar la operación realizada.</p>
--	---

Prototipos

Figura 1. Buscar Usuario:



Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Quedarán asignados los permisos a los roles seleccionados.

Tabla 14 Descripción de la funcionalidad "Eliminar Permiso"

Asignar Rol a Usuario:



Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador. Los roles deben estar previamente creados.
Funcionalidades tratadas	RF6, RF6.1, RF6.2
Conceptos tratados	Rol, Usuario
Descripción básica	<p>1 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción "Asignar Rol a Usuario" el sistema muestra la interfaz "Asignar Rol a Usuario" donde se visualizará un listado con los pares usuario-rol que existan en el sistema (ver figura 1). En caso de no existir ningún rol, el listado se mostrará vacío, con un mensaje "No se han asignado roles, Adicione roles a usuarios"</p> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Asignar</li> <li>-Eliminar</li> </ul> <p>2.1 Si el usuario selecciona la opción "Asignar" se mostrará la interfaz "Asignar Rol" (ver figura 2) donde podrá llenar un campo de texto con el usuario deseado y escoger el rol desde un listado despegable de roles existentes en el sistema:</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>-Usuario</li> <li>-Rol</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Salvar</li> <li>-Cancelar</li> </ul> <p>2.1.2 Si se da clic en el botón "Asignar" se adiciona el par usuario-rol a la lista. Se cerrará la ventana "Añadir Rol" mostrándose en la lista el nuevo par.</p> <p>2.1.3 Si se da clic en el botón "Cancelar" se cerrará la ventana "Asignar Usuario".</p> <p>2.3 Si el usuario selecciona la opción "Eliminar" se mostrará una ventana de confirmación (ver figura 3), si se confirma en el botón "Eliminar" se eliminará el par usuario-rol seleccionado y desaparecerá de la lista de roles. Si se cancela la operación con el botón "Cancelar" se cerrará la ventana de confirmación.</p>
Prototipos	

Figura 1. Asignar Rol a Usuario:

Asignar Rol a un Usuario

**Listado de Usuarios con sus Roles asignados**

Nombre del Usuario	Rol asignado al Usuario	
aecarrasco	Captación	
aecarrasco	Entrega	
aecarrasco	Supervisor	
dbarrera	Supervisor	
ldiaz	Administrador	
yalbelo	Administrador	

1 Mostrando 1 a 4 de 4 registros + Adicionar

Figura 2. Asignar Rol:

Asignar Rol a un Usuario

**Listado de Usuarios con sus Roles asignados**

Nombre del Usuario	Rol asignado al Usuario	
aecarrasco		
aecarrasco		
aecarrasco		
dbarrera		
ldiaz		
yalbelo		

1 Mostrando 1 a 4 de 4 registros + Adicionar

**+ Adicionar** x

Nombre del Usuario

Rol asignado al Usuario  
Administrador ▼

Figura 3. Eliminar Asignación:



Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Se eliminarán los pares roles-usuarios del sistema.

**Tabla 15 Descripción de la funcionalidad "Asignar Rol a Usuario"**

Gestionar Oficina:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador.
Funcionalidades tratadas	RF7, RF7.1, RF7.2, RF7.3
Conceptos tratados	Oficina
Descripción básica	<p>2 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul>

	<p>2 Si el usuario selecciona la opción "Gestionar Oficina" el sistema muestra la interfaz "Gestionar Oficina" (ver figura 1) donde se visualizará un listado con las oficinas que se han añadido al sistema, su estado y la cantidad de puestos de trabajo que las mismas poseen. En caso de no existir ninguna oficina, el listado se mostrará vacío, con un mensaje "No hay oficinas, Adicione oficinas al sistema"</p> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Adicionar</li> <li>-Modificar</li> <li>-Mostrar Información</li> </ul> <p>2.1 Si el usuario selecciona la opción "Adicionar" se mostrará la interfaz "Añadir Oficina" (ver figura 2) con los siguientes parámetros a llenar:</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>-Nombre</li> <li>-Cantidad de Puestos de Trabajo</li> <li>-Estado</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Salvar</li> <li>-Cancelar</li> </ul> <p>2.1.2 Si se da clic en el botón "Salvar" se adiciona la oficina a la lista de oficina y a la base de datos del sistema. Se cerrará la ventana "Añadir Oficina" y se mostrarán los datos de la oficina añadida en la lista de oficinas.</p> <p>2.1.3 Si se da clic en el botón "Cancelar" se cerrará la ventana "Añadir Oficina".</p> <p>2.2 Si el usuario selecciona la opción "Modificar" se mostrará la interfaz "Modificar Oficina" (ver figura 3) con los siguientes parámetros a llenar:</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>-Cantidad de Puestos de Trabajo</li> <li>-Estado</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Salvar</li> <li>-Cancelar</li> </ul> <p>2.2.1 Si se da clic en el botón "Salvar" se actualizan en el listado de oficina los datos modificados y en la base de datos del sistema. Se cerrará la ventana "Modificar Oficina".</p> <p>2.2.3 Si se da clic en el botón "Cancelar" se cerrará la ventana "Modificar Oficina".</p> <p>2.3 Si el usuario selecciona la opción "Mostrar información" se desplegará de la oficina seleccionada el listado de puestos de trabajo pertenecientes a la oficina seleccionada con sus respectivos datos. (Ver figura 4)</p>
Prototipos	

Figura 1. Gestionar Oficina:



Figura 2. Añadir Oficina:



Figura 3. Modificar Oficina:



Figura 4. Mostrar Información:

Listado de Oficinas			
Nombre	Cantidad de Puestos	Estado	
asd	1	Activo	
Oficina Central	2	Activo	
Oficina1	3	Activo	

Puestos de Trabajo de - Oficina1		
Nombre del Puesto	Usuario asignado al Puesto	Estado
Puesto1	aecarrasco	Activo
Puesto1	dbarrera	Activo

Descripción alterna:	
Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Se adicionarán y/o eliminarán las oficinas al sistema.

Tabla 16 Descripción de la funcionalidad "Gestionar Oficina"

Gestionar Puestos de Trabajo:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador. Las oficinas deben estar previamente creadas.
Funcionalidades tratadas	RF8, RF8.1, RF8.1.2, RF8.2, RF8.3
Conceptos tratados	Puesto de Trabajo, Oficina



Descripción básica	<p>1 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción " Puestos y Dispositivos" el sistema muestra la interfaz "Puestos y dispositivos"(ver figura 1) donde se visualizará un listado con los puestos que se han añadido al sistema. En caso de no existir ningún puesto, el listado se mostrará vacío, con un mensaje "No se han asignado puestos de trabajo, Adicione puestos de trabajo".</p> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Adicionar Puesto</li> <li>-Modificar Estado</li> </ul> <p>2.1 Si el usuario selecciona la opción "Adicionar Puesto" se mostrará la interfaz "Adicionar Puestos de Trabajo" (ver figura 3) con los siguientes parámetros a llenar:</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>-Puesto de rabajo</li> <li>-Tipo de puesto</li> <li>-Oficina a la que pertenece</li> <li>-Usuario asignado</li> <li>-MAC</li> <li>-Estado</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Salvar</li> <li>-Cancelar</li> </ul> <p>2.1.2 Si se da clic en el botón "Salvar" se adiciona el puesto a la lista. Se cerrará la ventana "Adicionar Puestos de Trabajo" mostrándose la lista con el nuevo puesto.</p> <p>2.1.3 Si se da clic en el botón "Cancelar" se cerrará la ventana "Adicionar Puesto de Trabajo".</p> <p>2.2 Si el usuario selecciona la opción "Modificar Puesto" se mostrará la interfaz "Modificar Puestos de Trabajo" con los siguientes parámetros a llenar:</p> <p><b>Parámetros a llenar:</b></p> <ul style="list-style-type: none"> <li>-Estado</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Salvar</li> <li>-Cancelar</li> </ul> <p>2.2.1 Si se da clic en el botón "Salvar" se modifica el par en la lista. Se cerrará</p>
--------------------	--

la ventana "Modificar Puesto de Trabajo" y se mostrará los cambios realizados.  
2.2.3 Si se da clic en el botón "Cancelar" se cerrará la ventana "Modificar Puesto de Trabajo".

## Prototipos

Figura 1. Gestionar Puestos de Trabajo:

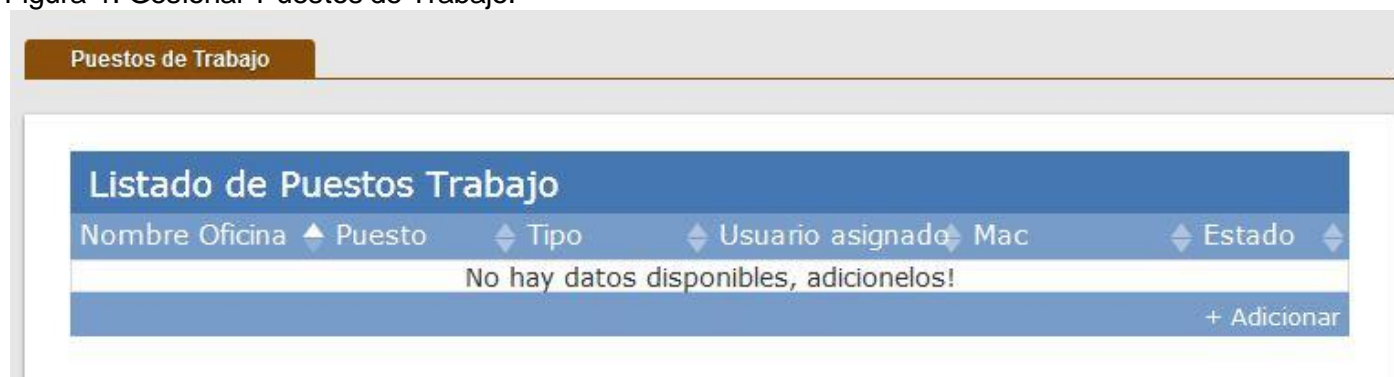


Figura 2. Adicionar Puesto :



Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Quedarán añadidos los puestos de trabajo.

**Tabla 17 Descripción de la funcionalidad "Gestionar Puestos de Trabajo"**

Configurar Aplicación:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador.
Funcionalidades tratadas	RF9, RF9.1, RF9.2
Conceptos tratados	Módulo

Descripción básica	<p>1 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción "Configurar Aplicación" el sistema muestra la interfaz "Configurar Aplicación" (ver figura 1) donde se visualizarán los módulos existentes brindando la opción de habilitarlos y/o deshabilitarlos.</p> <p>Se brindará además la opción de cambiar el tema de las interfaces de la aplicación.</p> <p><b>Acciones:</b></p> <p>-Guardar</p> <p>2.1 Si el usuario luego de realizar dichas operaciones, da clic en el botón "Guardar" los cambios serán salvados.</p>
Prototipos	

Figura 1 Configurar Aplicación:

## Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Quedarán activados solo los módulos seleccionados por el usuario administrador. El tema de las interfaces quedará configurado de la manera seleccionada por el administrador.

Tabla 18 Descripción de la funcionalidad "Configurar Aplicación"

## Registro de Operaciones:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador.
Funcionalidades tratadas	RF10, RF10.1, RF10.2
Conceptos tratados	Módulo
Descripción básica	<p>1 El sistema muestra un menú lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar Ciudadano</li> <li>• Gestionar Usuario</li> <li>• Gestionar Oficina</li> <li>• Gestionar Puestos</li> <li>• Gestionar Rol</li> <li>• Asignar Rol a Usuario</li> <li>• Asignar Permisos</li> <li>• Eliminar Permisos</li> <li>• Nomencladores</li> <li>• Registros de Operaciones</li> </ul>

	<ul style="list-style-type: none"> <li>• Configurar Aplicación</li> <li>• Logs</li> </ul> <p>2 Si el usuario selecciona la opción "Registros de Operaciones" el sistema muestra la interfaz " Registros de Operaciones" (ver figura 1) donde se mostrarán los siguientes campos a llenar:</p> <p><b>Campos a llenar:</b></p> <ul style="list-style-type: none"> <li>-Fecha</li> <li>-Acción</li> <li>-Usuario</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Filtrar</li> </ul> <p>2.1 Si el usuario luego de introducir los datos da clic en el botón "Filtrar" se mostrará un registro de la fecha en la que dicho usuario ha realizado tales acciones, además de una descripción de las mismas.</p>
--	--

Prototipos

Figura 1 Registro de Operaciones

The screenshot shows a web interface titled "Operaciones de los Usuarios". At the top, there is a search bar with three input fields: "Fecha:", "Acción", and "Usuario", followed by a blue "Filtrar" button. Below the search bar is a table titled "Lista de Operaciones". The table has four columns: "Acción", "Usuario", "Fecha", and "Descripción". It contains five rows of data, all with the date "14-06-2013". At the bottom of the table, there is a status bar that says "1 Mostrando 1 a 5 de 5 registros".

Acción	Usuario	Fecha	Descripción
Adicionar	ldiaz	14-06-2013	Adicionó un puesto de trabajo en la Oficina de nombre: Oficina1
Adicionar	ldiaz	14-06-2013	Adicionó el Rol: Administrador al Usuario: yalbelo
Adicionar	ldiaz	14-06-2013	Adicionó un puesto de trabajo en la Oficina de nombre: Oficina1
Adicionar	ldiaz	14-06-2013	Adicionó un puesto de trabajo en la Oficina de nombre: Oficina1
Adicionar	ldiaz	14-06-2013	Adicionó el Rol: Supervisor al Usuario: aecarrasco

Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	Se registrarán las operaciones realizadas por los usuarios de forma detallada.

Tabla 19 Descripción de la funcionalidad "Registro de Operaciones"

Editar Perfil de Usuario del Sistema:

Precondiciones	El usuario debe estar autenticado en el sistema con el rol administrador.
Funcionalidades tratadas	RF11, RF11.1
Conceptos tratados	Módulo
Descripción básica	<p>1 El sistema muestra en la parte superior derecha un botón para la desconexión y otro para cambiar el perfil de usuario. (Ver figura 1)</p> <p>2 Si el usuario selecciona la opción " Editar Perfil de Usuario del Sistema" el sistema muestra la interfaz "Registros de Operaciones" (ver figura 2) donde se mostrarán los siguientes campos a llenar:</p> <p><b>Campos a llenar:</b></p> <ul style="list-style-type: none"> <li>-Contraseña anterior</li> <li>-Nueva contraseña</li> <li>-Confirmar Contraseña</li> </ul> <p><b>Acciones:</b></p> <ul style="list-style-type: none"> <li>-Guardar</li> </ul> <p>2.1 Si el usuario luego de introducir los datos da clic en el botón "Guardar" guardará la nueva contraseña.</p>

Prototipos

Figura 1. Panel Superior Derecho:

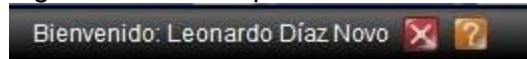


Figura 2. Editar Perfil de Usuario:

Descripción alterna:

Validaciones	Los datos introducidos por el usuario deben estar correctos.
Postcondiciones	La nueva contraseña se utilizará para entrar en el sistema por el dominio local.

Tabla 20 Descripción de la funcionalidad "Editar Perfil de Usuario del Sistema"

### ANEXO 3. Diagramas de clases del diseño

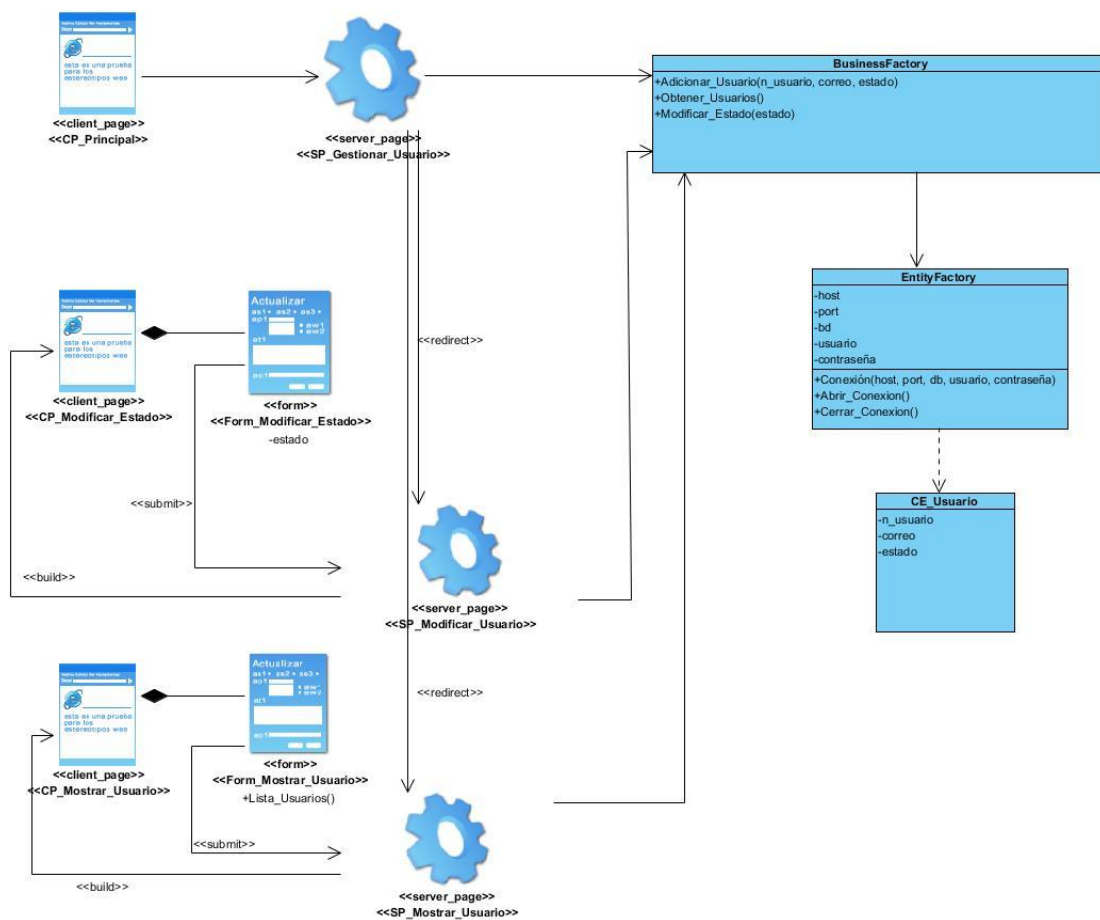


Figura 21 Diagrama de clases del diseño "Gestionar Usuario"

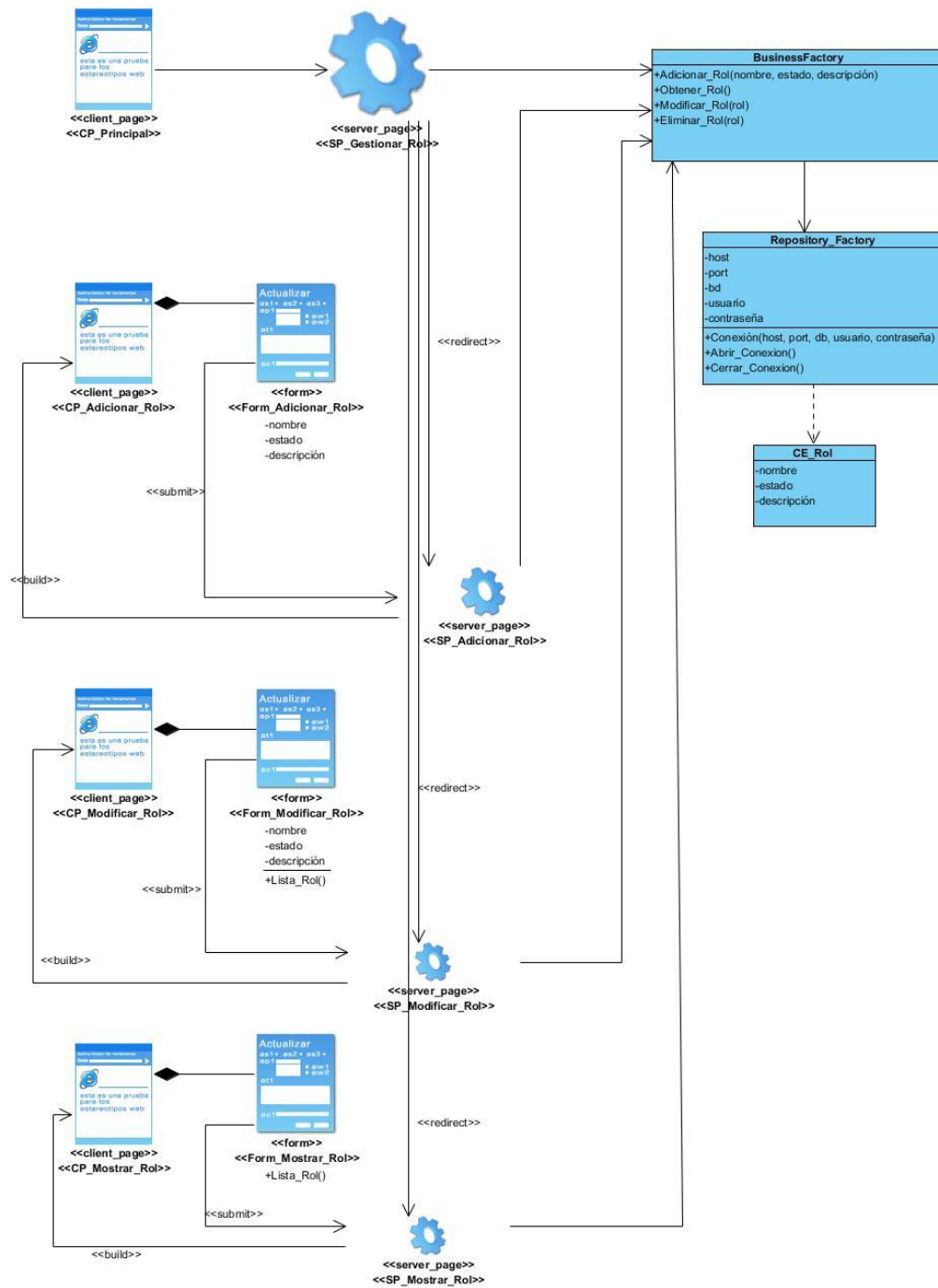


Figura 22 Diagrama de clases del diseño "Gestionar Rol"



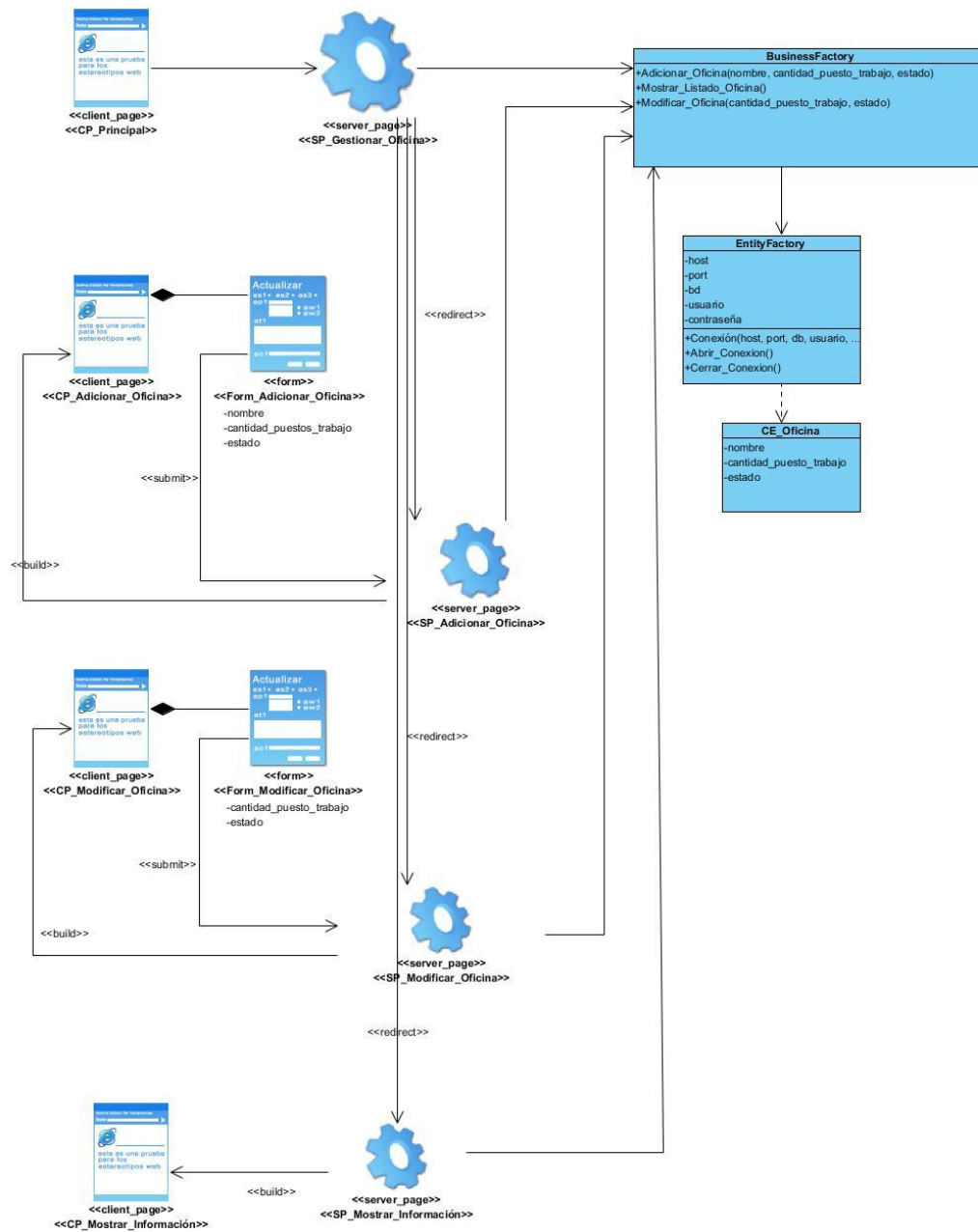


Figura 23 Diagrama de clases del diseño "Gestionar Oficinas"



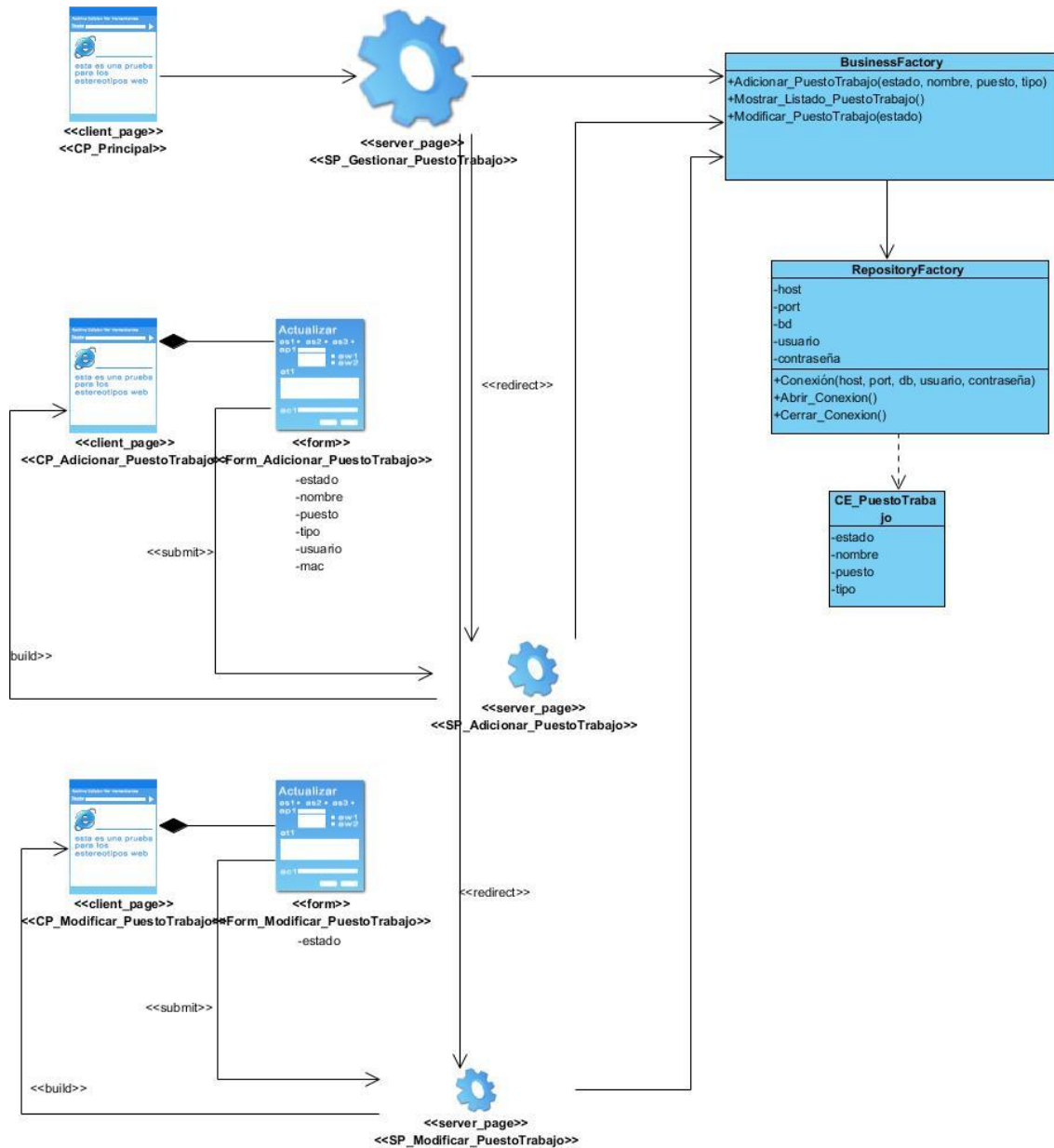


Figura 24 Diagrama de clases del diseño "Gestionar Puestos de Trabajo"

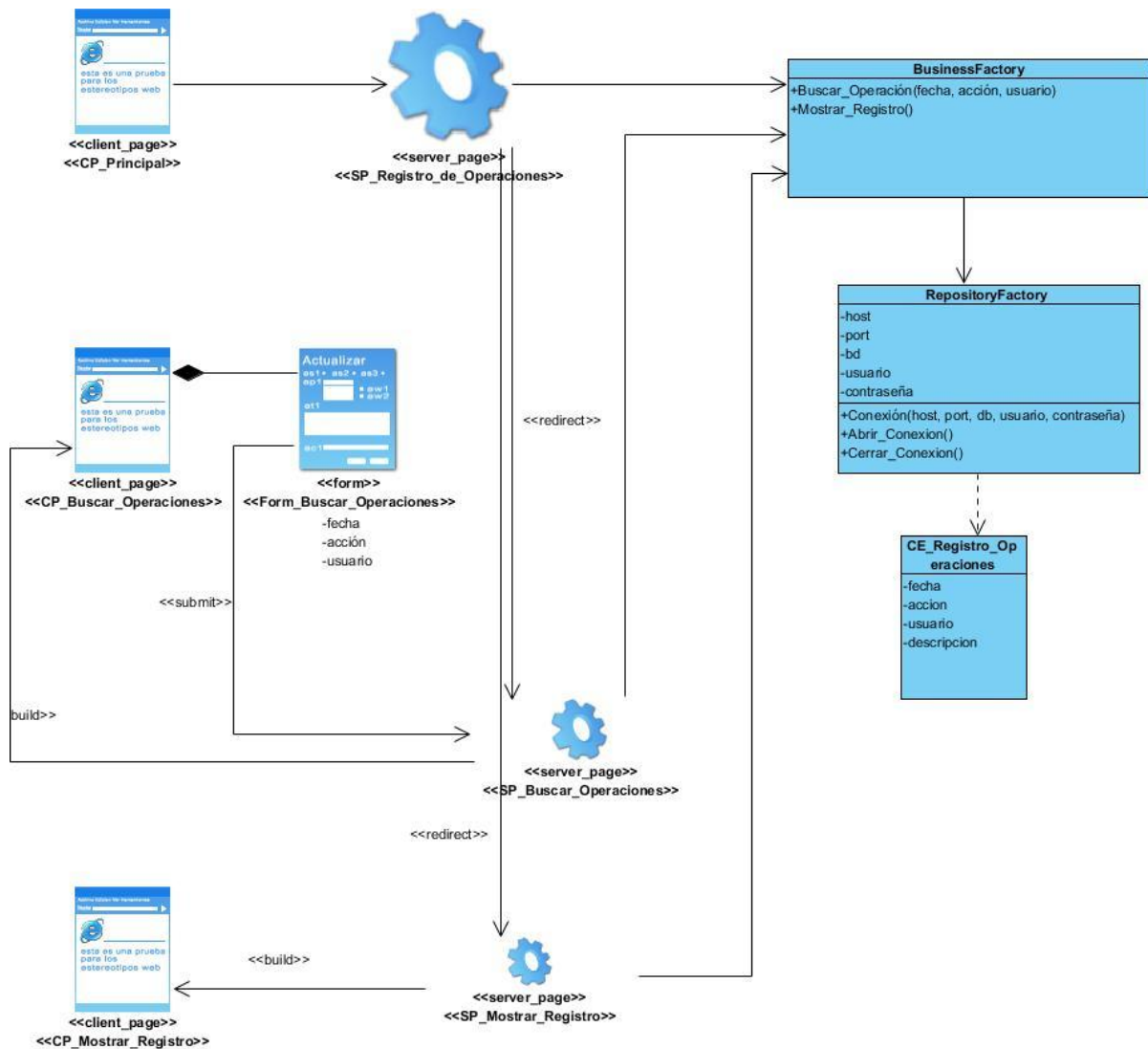


Figura 25 Diagrama de clases del diseño Registro de Operaciones

## ANEXO 4. Descripción de las entidades del Modelo de Datos

Nombre de la tabla	dCiudadano		
Atributos	Tipo de Datos	Nulo	Definición
id_ciudadano (PK)	uuid	no	Identificador del ciudadano
primer_nombre	varchar(25)	no	Primer nombre del ciudadano

segundo_nombre	varchar(25)	si	Segundo nombre del ciudadano
primer_apellido	varchar(25)	no	Primer apellido del ciudadano
segundo_apellido	varchar(25)	no	Segundo apellido del ciudadano
ci	varchar(11)		Carnet de identidad del ciudadano
sexo	char	no	Sexo del ciudadano
fecha_Nacimiento	timestamp	no	Fecha de nacimiento del ciudadano

*Tabla 21 Entidad correspondiente al ciudadano*

<b>Nombre de la tabla</b>	nTipo_Ciudadano		
<b>Atributos</b>	<b>Tipo de Datos</b>	<b>Nulo</b>	<b>Definición</b>
id_ciudadano (PK)	uuid	no	Identificador del tipo de ciudadano
clasificación	varchar(25)	no	Clasificación del tipo de ciudadano

*Tabla 22 Entidad correspondiente al tipo de ciudadano*

<b>Nombre de la tabla</b>	dUsuario		
<b>Atributos</b>	<b>Tipo de Datos</b>	<b>Nulo</b>	<b>Definición</b>
id_usuario (PK)	uuid	no	Identificador del ciudadano
estado	bool	si	Estado del usuario
correo	varchar(25)	no	Correo del usuario
contraseña	varchar(25)	no	Contraseña del usuario
nombre_usuario	varchar(25)	si	Nombre del usuario
id_ciudadano (FK)	uuid	no	Identificador de la tabla <i>dCiudadano</i>

*Tabla 23 Entidad correspondiente al usuario*

Nombre de la tabla	dRol		
Atributos	Tipo de Datos	Nulo	Definición
id_rol (PK)	uuid	no	Identificador del rol
estado	bool	no	Estado del rol
nombre_rol	varchar(25)	no	Nombre del rol
descripción	varchar(25)	si	Descripción del rol

*Tabla 24 Entidad correspondiente al rol*

Nombre de la tabla	dUsuario_Rol		
Atributos	Tipo de Datos	Nulo	Definición
id_rol (PK) (FK)	uuid	no	Identificador de la tabla <i>dRol</i>
id_usuario(PK) (FK)	uuid	no	Identificador de la tabla <i>dUsuario</i>

*Tabla 25 Entidad correspondiente a la relación usuario-rol*

Nombre de la tabla	dPermiso		
Atributos	Tipo de Datos	Nulo	Definición
id_permiso (PK)	uuid	no	Identificador del permiso
dirección	varchar(25)	si	Dirección perteneciente a los permisos
nombre	varchar(25)	no	Estado de la oficina
id_rol (FK)	uuid	no	Identificador de la tabla <i>nRol</i>

*Tabla 26 Entidad correspondiente a los permisos*

Nombre de la tabla	dOficina		
Atributos	Tipo de Datos	Nulo	Definición
id_oficina (PK)	uuid	no	Identificador de la oficina
cantidad_puesto_trabajo	integer(14)	no	Cantidad de puestos de trabajo en la oficina
descripción	varchar(25)	si	Descripción de la oficina
estado	bool	no	Estado de la oficina

id_tipo_oficina (FK)	uuid	no	Identificador de la tabla <i>ntipo_oficina</i>
----------------------	------	----	--

Tabla 27 Entidad correspondiente a las oficinas

Nombre de la tabla	nTipo_Puesto		
Atributos	Tipo de Datos	Nulo	Definición
id_tipo_puesto (PK)	uuid	no	Identificador del tipo de puesto
nombre	varchar(25)	no	nombre del tipo de puesto

Tabla 28 Entidad correspondiente a los tipos de puesto

Nombre de la tabla	dPuesto_Trabajo		
Atributos	Tipo de Datos	Nulo	Definición
id_puesto_trabajo (PK)	uuid	no	Identificador del puesto de trabajo
nombre	varchar(25)	no	Nombre del puesto de trabajo
mac	varchar(17)	no	MAC del puesto de trabajo
estado	bool	no	Estado del puesto de trabajo
id_oficina (FK)	uuid	no	Identificador de la tabla <i>dOficina</i>
id_tipo_de_puesto (FK)	uuid	no	Identificador de la tabla <i>dTipo_Puesto</i>
id_usuario (FK)	uuid	no	Identificador de la tabla <i>dUsuario</i>

Tabla 29 Entidad correspondiente a los puestos de trabajo

Nombre de la tabla	dReporte_Operaciones		
Atributos	Tipo de Datos	Nulo	Definición
id_reporte (PK)	uuid	no	Identificador del reporte
fecha	timestamp	no	Fecha de las operaciones
descripción	archar(25)	no	Descripción de las operaciones
id_usuario (FK)	uuid	no	Identificador de la tabla <i>dUsuario</i>
id_tipo_acción (FK)	uuid	no	Identificador de la tabla

			<i>dTipo_Acción</i>
--	--	--	---------------------

Tabla 30 Entidad correspondiente a los reportes de operaciones

Nombre de la tabla	dTipo_Acción		
Atributos	Tipo de Datos	Nulo	Definición
id_tipo_acción (PK)	uuid	no	Identificador del tipo de acción
nombre	varchar(25)	no	Nombre del tipo de acción

Tabla 31 Entidad correspondiente al tipo de acción

## ANEXO 5. Diagramas de secuencia

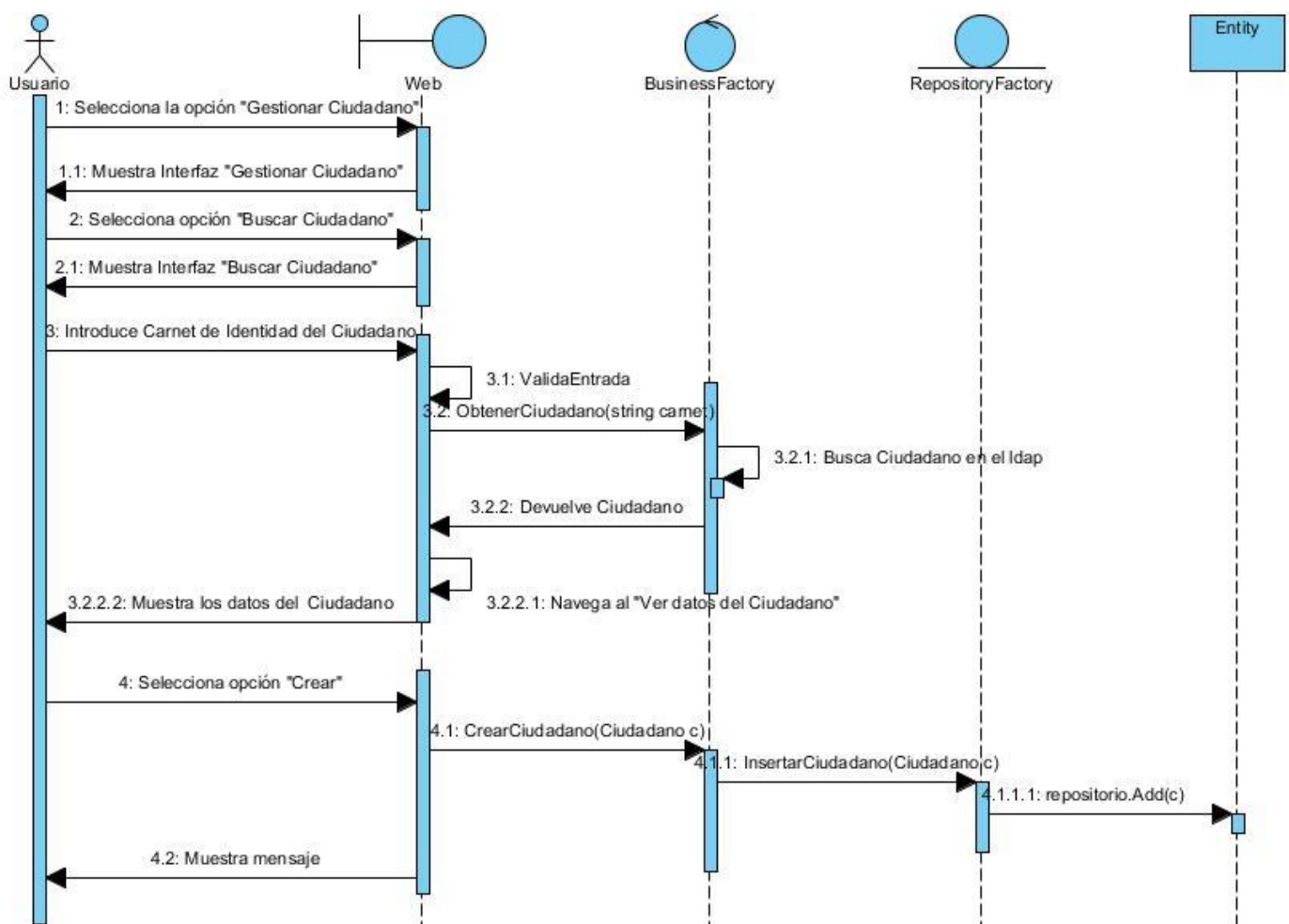
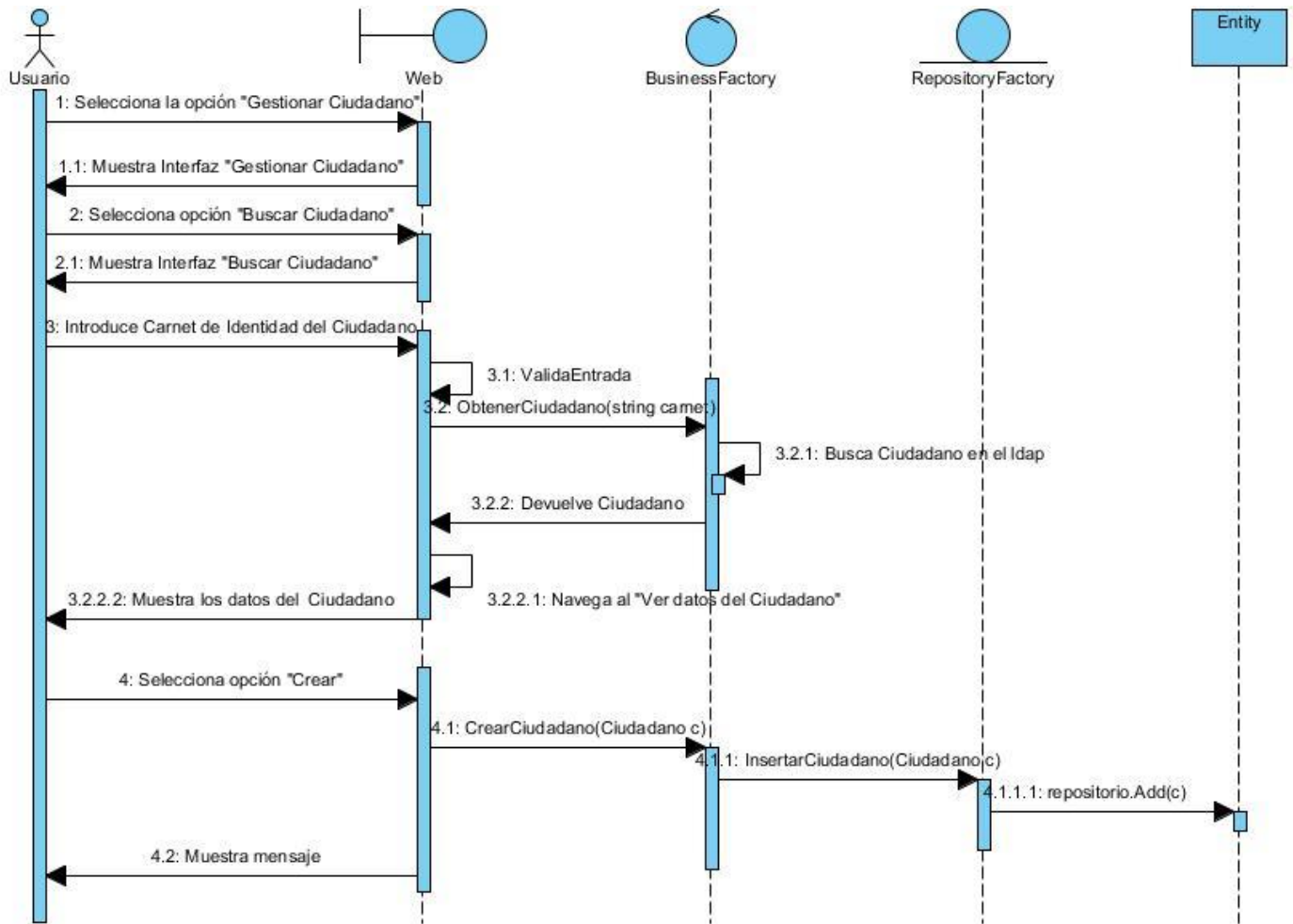
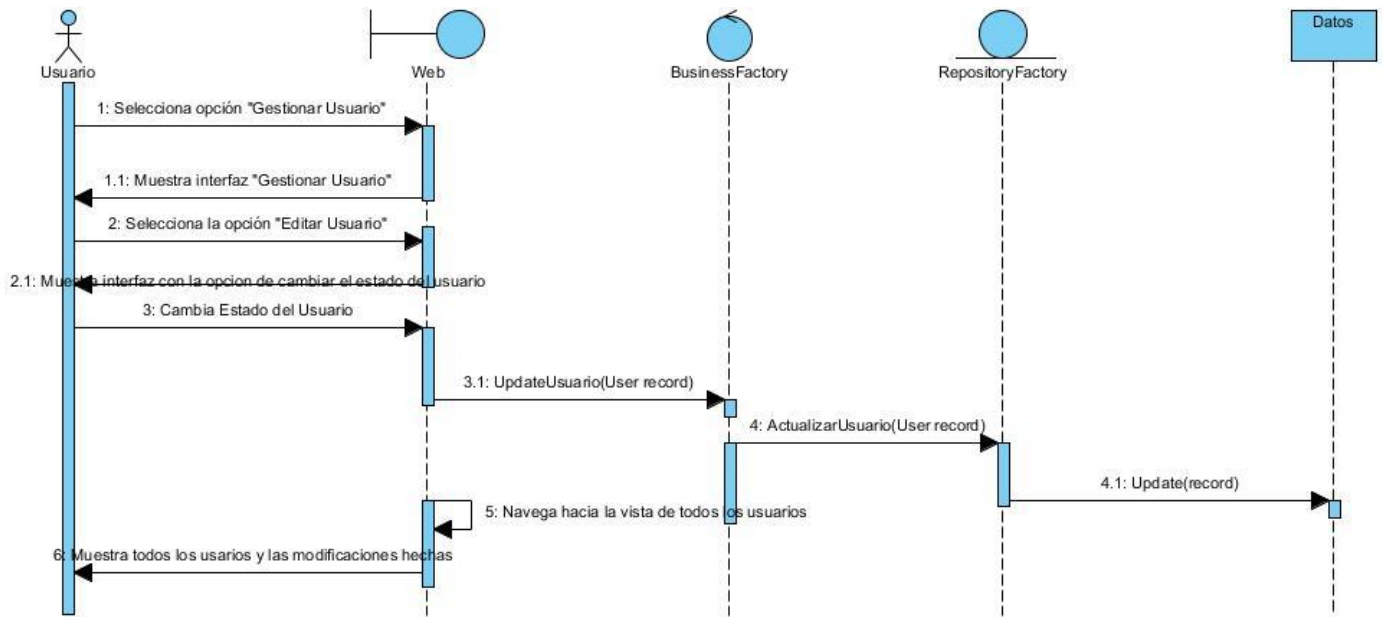


Figura 26 Diagrama de secuencia "Gestionar Ciudadano". Escenario "Buscar Ciudadano"

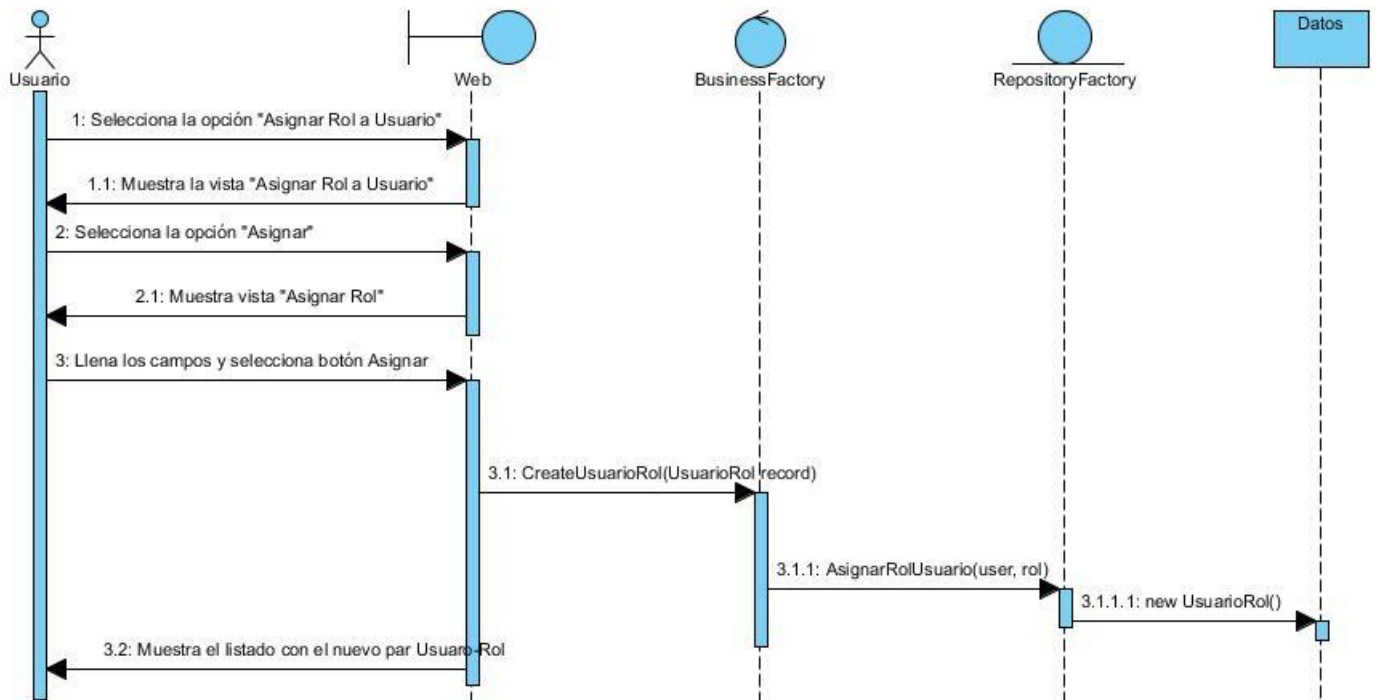


**Figura 27 Diagrama de secuencia "Gestionar Ciudadano". Escenario "Crear Ciudadano"**





**Figura 28 Diagrama de secuencia "Gestionar Usuario". Escenario "Editar Usuario"**



**Figura 29 Diagrama de secuencia "Asignar Rol a Usuario"**

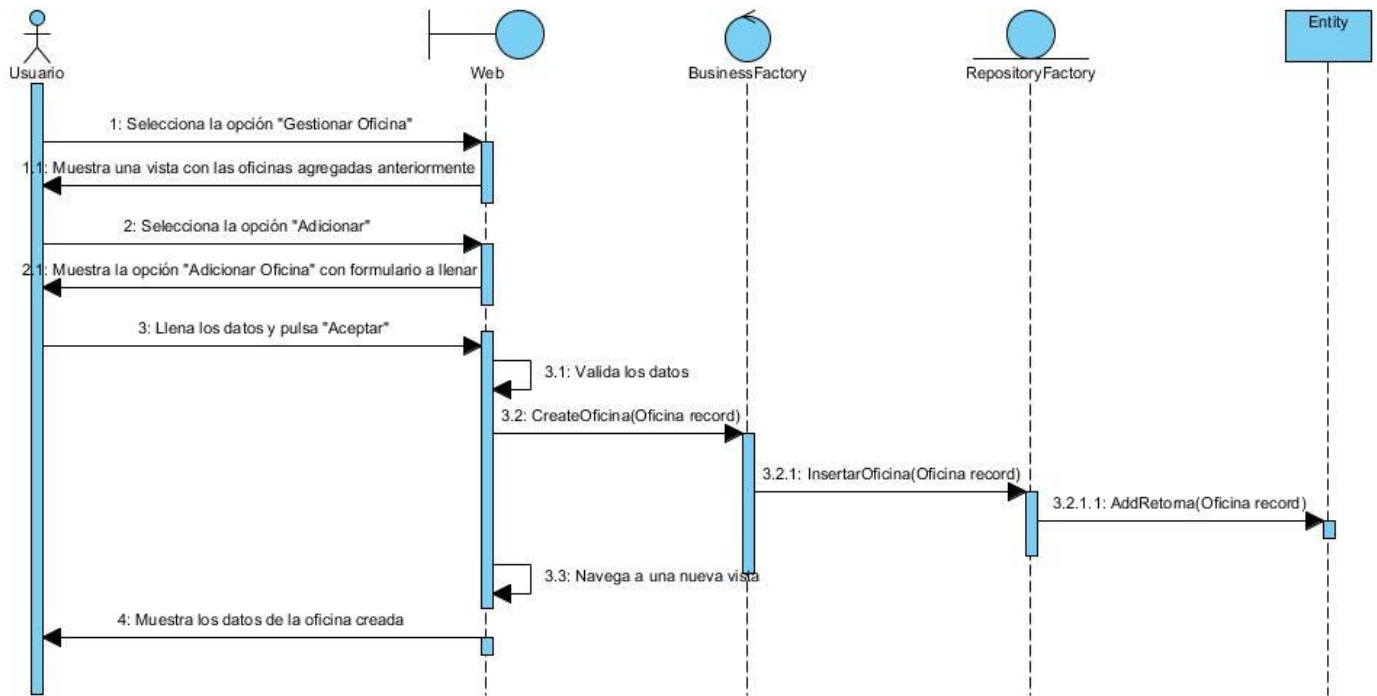


Figura 30 Diagrama de secuencia "Gestionar Oficina". Escenario "Crear Oficina"

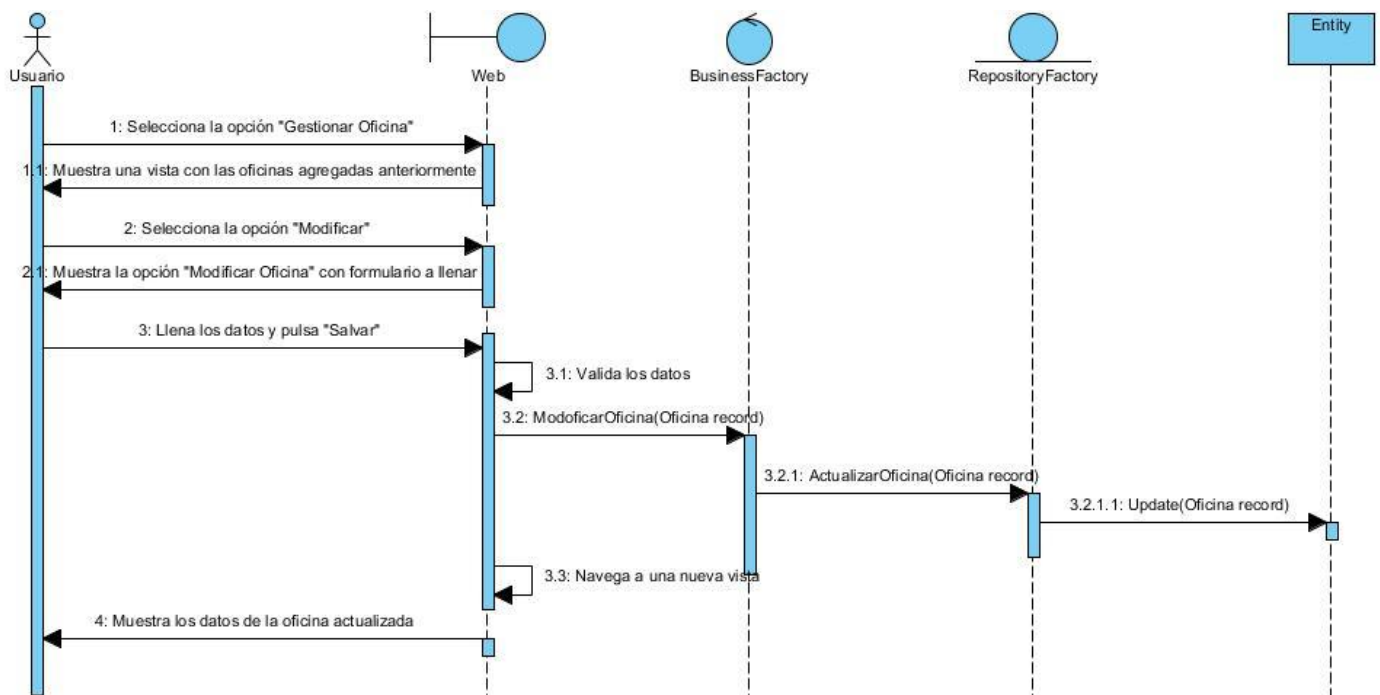


Figura 31 Diagrama de secuencia "Gestionar Oficina". Escenario "Actualizar Oficina"

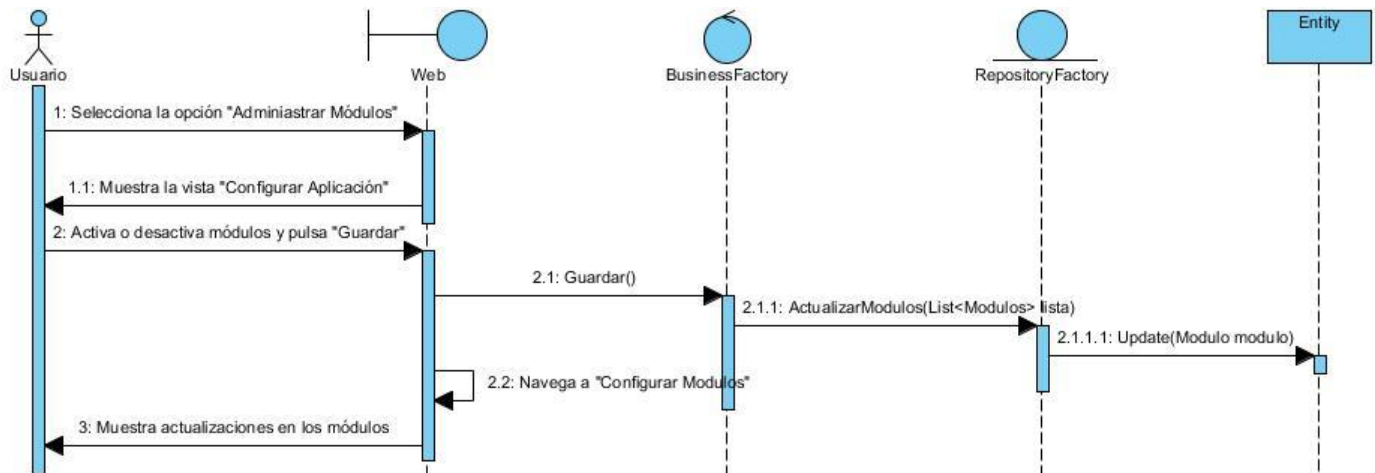


Figura 32 Diagrama de secuencia "Configurar Aplicación"

## ANEXO 6. Casos de prueba de caja negra

### CP Gestionar Ciudadano:

#### SC1: Buscar Ciudadano

Escenario	Descripción	CI	Solapamiento	Respuesta del sistema	Flujo central
EC 1.1 Se realiza la búsqueda satisfactoriamente.	El usuario introduce el carnet de identidad del ciudadano a encontrar.	V	V	El sistema realiza una búsqueda según el criterio entrado por el usuario.	1-El usuario selecciona la opción "Gestionar Ciudadano". 2-El sistema muestra la interfaz "Gestionar Ciudadano". 3-Entra los datos solicitados. 4-Da clic en el botón "Buscar".

EC 1.2 Campos Vacíos.	El usuario deja de insertar algún dato.	V	NA	Se muestra un mensaje al lado del campo sin llenar con el siguiente texto: "Debe introducir los datos requeridos".	<p>1-El usuario selecciona la opción "Gestionar Ciudadano".</p> <p>2-El sistema muestra la interfaz "Gestionar Ciudadano".</p> <p>3-Entra los datos solicitados.</p> <p>4-Da clic en el botón "Buscar".</p>
EC 1.3 Caracteres Extraños	El usuario introduce datos con caracteres extraños.	I	V	Se muestra un mensaje al lado del campo incorrecto: "Debe introducir los datos correctamente".	<p>1-El usuario selecciona la opción "Gestionar Oficina".</p> <p>2-El sistema muestra la interfaz "Gestionar Oficina".</p> <p>3-Selecciona la opción "Añadir Oficina"</p> <p>4-Entra los datos solicitados.</p> <p>5-Da clic en el botón "Salvar".</p>

Tabla 32 Escenario "Buscar Ciudadano" del caso de prueba "Gestionar Ciudadano"

## SC2: Crear Ciudadano

Escenario	Descripción	CI	Solapín	Respuesta del sistema	Flujo central
-----------	-------------	----	---------	-----------------------	---------------

<i>EC 2.1 Se realiza una búsqueda satisfactoriamente.</i>	<i>El usuario introduce el carnet de identidad del ciudadano a encontrar.</i>	V	V	<i>El sistema realiza una búsqueda según el criterio entrado por el usuario.</i>	<p>1-El usuario selecciona la opción "Gestionar Ciudadano".</p> <p>2-El sistema muestra la interfaz "Gestionar Ciudadano".</p> <p>3-Entra los datos solicitados.</p> <p>4-Se muestran los datos del ciudadano buscado.</p> <p>5-Da clic en el botón "Crear"</p>
---	---	---	---	--	---

**Tabla 33 Escenario "Crear Ciudadano" del caso de prueba "Gestionar Ciudadano"**

## CP Asignar Rol a Usuario:

### SC1: Asignar Rol

Escenario	Descripción	Nombre	Rol	Respuesta del sistema	Flujo central
<i>EC 1.1 Se asigna el rol satisfactoriamente.</i>	<i>El usuario introduce los datos del par usuario-rol a registrar.</i>	V	V	<i>El sistema asigna el rol al usuario seleccionado.</i>	<p>1-El usuario selecciona la opción "Asignar Rol a Usuario".</p> <p>2-El sistema muestra la interfaz "Asignar Rol a Usuario".</p> <p>3-Selecciona la opción "Adicionar"</p> <p>4-Entra los datos solicitados.</p> <p>5-Da clic en el botón "Salvar".</p>

<i>EC 1.2 Campos Vacíos.</i>	<i>El usuario deja de insertar algún dato.</i>	<i>NA</i>	<i>V</i>	<i>Se muestra un mensaje al lado del campo sin llenar con el siguiente texto: "Debe introducir los datos requeridos".</i>	<p><i>1-El usuario selecciona la opción "Asignar Rol a Usuario".</i></p> <p><i>2-El sistema muestra la interfaz "Asignar Rol a Usuario".</i></p> <p><i>3-Selecciona la opción "Adicionar".</i></p> <p><i>4-Entra los datos solicitados.</i></p> <p><i>5-Da clic en el botón "Salvar".</i></p>
<i>EC 1.3 Caracteres Extraños</i>	<i>El usuario introduce datos con caracteres extraños.</i>	<i>I</i>	<i>V</i>	<i>Se muestra un mensaje al lado del campo incorrecto: "Debe introducir los datos correctamente".</i>	<p><i>1-El usuario selecciona la opción "Asignar Rol a Usuario".</i></p> <p><i>2-El sistema muestra la interfaz "Asignar Rol a Usuario".</i></p> <p><i>3-Selecciona la opción "Adicionar".</i></p> <p><i>4-Entra los datos solicitados.</i></p> <p><i>5-Da clic en el botón "Salvar".</i></p>

*Tabla 34 Escenario "Asignar Rol" del caso de prueba "Asignar Rol a Usuario"*

## SC2: Eliminar Asignación

Escenario	Descripción	Nombre	Descripción	Estado	Respuesta del sistema	Flujo central
-----------	-------------	--------	-------------	--------	-----------------------	---------------

<i>EC 2.1 Se elimina la asignación seleccionada satisfactoriamente.</i>	<i>El usuario selecciona esta opción para eliminar un par usuario-rol.</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>El sistema elimina el par usuario-rol seleccionado de la base de datos y de la lista de roles.</i>	<i>1-El usuario selecciona la opción "Asignar Rol a Usuario".</i> <i>2-El sistema muestra la interfaz "Asignar Rol a Usuario".</i> <i>3-Selecciona la opción "Eliminar"</i> <i>4-Confirma la selección dando clic en el botón "Salvar".</i>
---	--	----------	----------	----------	---	--

*Tabla 35 Escenario "Eliminar Asignación" del caso de prueba "Asignar Rol a Usuario"*

## CP Gestionar Oficina:

### SC1: Añadir Oficina

Escenario	Descripción	Nombre	Cantidad de Puestos	Estado	Respuesta del sistema	Flujo central
-----------	-------------	--------	---------------------	--------	-----------------------	---------------



EC 1.1 Se añade la oficina satisfactoriamente.	El usuario introduce los datos de la oficina a registrar.	V	V	V	El sistema inserta la oficina con los datos especificados.	<p>1-El usuario selecciona la opción "Gestionar Oficina".</p> <p>2-El sistema muestra la interfaz "Gestionar Oficina".</p> <p>3-Selecciona la opción "Añadir Oficina"</p> <p>4-Entra los datos solicitados.</p> <p>5-Da clic en el botón "Salvar".</p>
EC 1.2 Campos Vacíos.	El usuario deja de insertar algún dato.	V	NA	V	Se muestra un mensaje al lado del campo sin llenar con el siguiente texto: "Debe introducir los datos requeridos".	<p>1-El usuario selecciona la opción "Gestionar Oficina".</p> <p>2-El sistema muestra la interfaz "Gestionar Oficina".</p> <p>3-Selecciona la opción "Añadir Oficina"</p> <p>4-Entra los datos solicitados.</p> <p>5-Da clic en el botón "Salvar".</p>
EC 1.3 Caracteres Extraños	El usuario introduce datos con caracteres	I	V	V	Se muestra un mensaje al lado del campo incorrecto: "Debe introducir los	1-El usuario selecciona la opción "Gestionar

	extraños.				datos correctamente”.	Oficina”. 2-El sistema muestra la interfaz “Gestionar Oficina”. 3-Selecciona la opción “Añadir Oficina” 4-Entra los datos solicitados. 5-Da clic en el botón “Salvar”.
--	-----------	--	--	--	-----------------------	---

Tabla 36 Escenario “Añadir Oficina” del caso de prueba “Gestionar Oficina”

## SC2: Modificar Oficina

Escenario	Descripción	Nombre	Cantidad de Puestos	Estado	Respuesta del sistema	Flujo central
EC 2.1 Se modifica la oficina satisfactoriamente.	El usuario modifica los datos de la oficina seleccionada.	V	V	V	El sistema actualiza los datos en la base de datos y en el listado de oficina.	1-El usuario selecciona la opción “Gestionar Oficina”. 2-El sistema muestra la interfaz “Gestionar Oficina”. 3-Selecciona la opción “Modificar Oficina” 4-Entra los datos solicitados. 5-Da clic en el botón “Salvar”.

Tabla 37 Escenario “Gestionar Oficina” del caso de prueba “Gestionar Oficina”

## ANEXO 7. Resultado de las pruebas de caja blanca

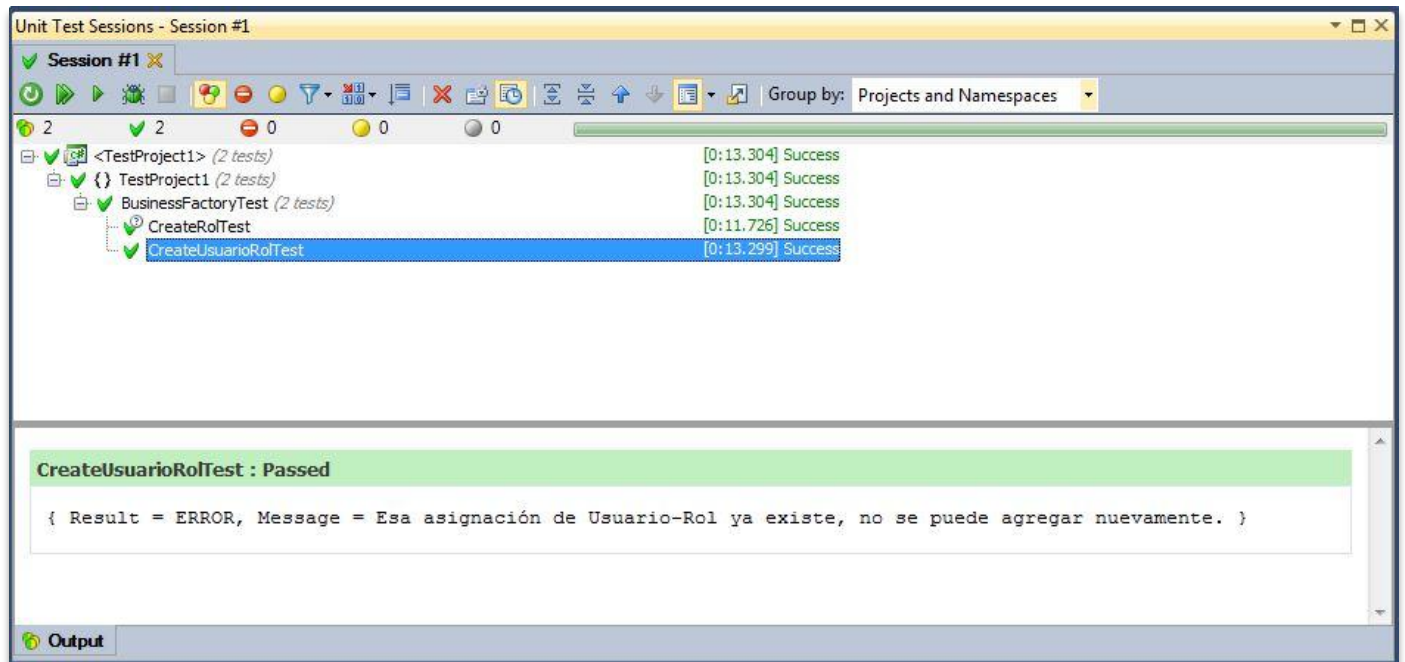
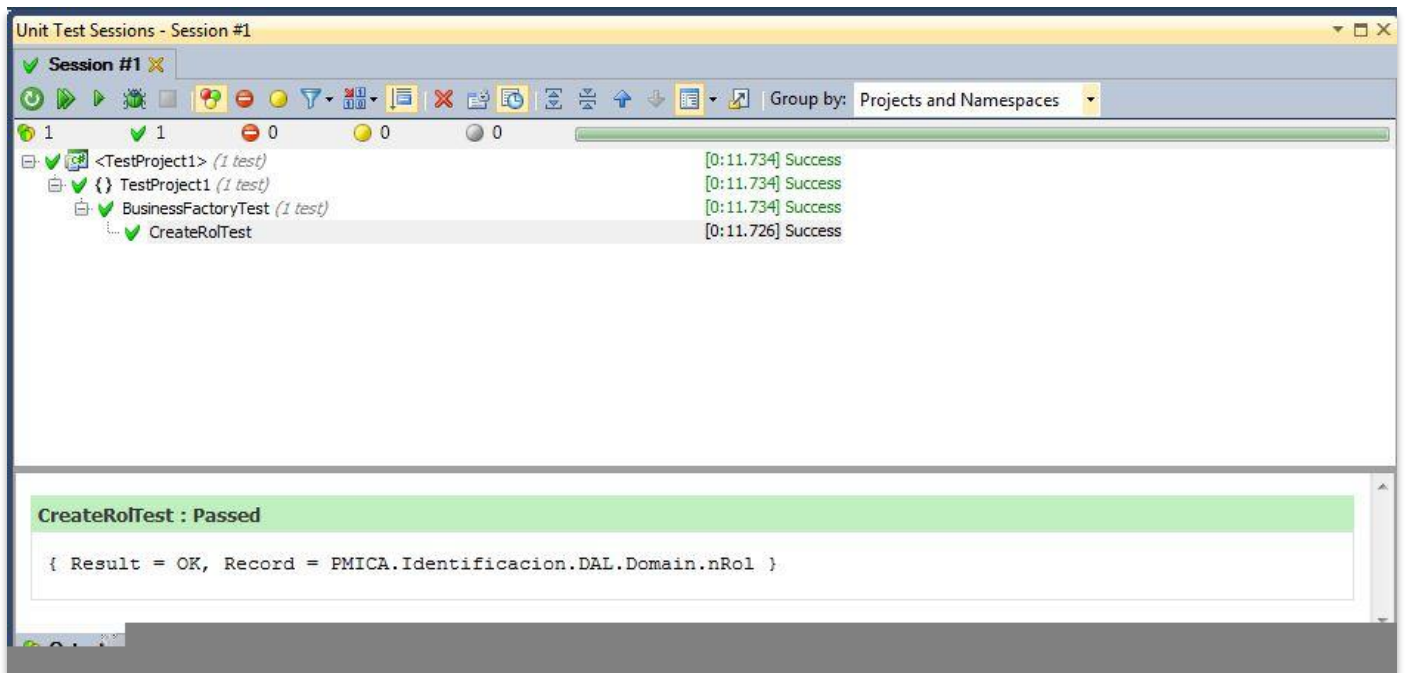


Figura 33 Resultado de la prueba unitaria realizada a la funcionalidad `CrearUsuarioRol()`



**Figura 34** Resultado de la prueba unitaria realizada a la funcionalidad `CrearRol()`