



**FACULTAD 4**

**SISTEMA DE CONTROL DE ACCESO BASADO EN IDENTIFICACIÓN POR  
RADIOFRECUENCIA.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN  
CIENCIAS INFORMÁTICAS**

**AUTOR:** Yasiel Silva Gómez

**TUTORES:** Ing. Diana Tahirí Galí Ramírez

Ing. Frank Geiler Vega Duvergel

**Co Tutor:** Ing. Yubismel Perdomo Velázquez

La Habana, 2018

Pensamiento



Si tú no construyes tu sueño, alguien va a contratarte para que le ayudes a construir el suyo.

Dhirubhai Ambani

## **Declaración de autoría**

Declaro ser el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, firmo el presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Yasiel Silva Gómez

Firma de autor

---

Ing. Diana Tahirí Galí Ramírez

Firma del tutor

---

Ing. Frank Geiler Vega Duvergel

Firma del tutor

---

Ing. Yubismel Perdomo Velázquez

Firma del tutor

## **Datos de contacto**

### **Autor:**

Yasiel Silva Gómez.

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [ysilva@estudiantes.uci.cu](mailto:ysilva@estudiantes.uci.cu)

### **Tutor:**

Diana Tahirí Galí Ramírez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [dtgali@uci.cu](mailto:dtgali@uci.cu)

### **Tutor:**

Frank Geiler Vega Duvergel

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [fgvega@uci.cu](mailto:fgvega@uci.cu)

### **Tutor:**

Yubismel Perdomo Velázquez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [yubismel@uci.cu](mailto:yubismel@uci.cu)

## **Dedicatoria**

A mi papá, que desgraciadamente no está entre nosotros y sé que donde quiera que esté, estaría orgulloso del hombre que soy porque he logrado lo que él siempre quiso. Si llegué a donde estoy ahora, fue pensando en él.

A mi mamá que siempre luchó por mí, por ser una buena persona y no dejar que nunca me faltara de nada, por su amor y cariño incondicional.

Por último, a mi hermano Mayito, que puede que no entienda estas palabras porque no es una persona normal como los demás, pero sé que sabe que lo quiero con todo mi corazón al igual que sé que él me quiere a mí, por él es que he logrado todo esto porque yo soy su futuro y siempre me he preocupado por él como si fuera más que un hermano un hijo, aunque sea mayor que yo, te quiero mucho gordito.

## Agradecimientos

A mis padres Mario y Nidia por su apoyo incondicional, por darme todo sin pedir nada a cambio, por darme todo lo que necesitaba, aunque las cosas no estuvieran a su alcance, por brindarme su amor, por hacer de mí un buen hijo y una buena persona. Mi papá que ya no está, pero a él le agradezco todo porque siempre confió en mí y sabía que yo podía llegar hasta donde llegué, que hubiese querido que escuchara estas palabras, pero la vida nos da muchos golpes de los cuales hay que defenderse.

A mi tía Miriam pues sin ella todo hubiera sido muy difícil para mí, gracias a ella nunca me hizo falta comprar ni ropa ni zapatos, gracias a ella fue que pude terminar la universidad después de la muerte de mi papá, gracias tía por hacer tanto por mí sin pedirme nada a cambio.

A mi primo Maikel que me ayudó en todo lo que pudo, por apoyarme en todo lo que hizo falta.

A mi tía Mare y Roberto que siempre me han ayudado en todo tanto material como emocionalmente, por todos los consejos que me daban desde chiquitico, por preocuparse por mí y mi hermano después de la muerte de mi papá y desde siempre porque siempre les hemos importado.

A mi hermana Yoandra, que siempre me ha apoyado y querido mucho, por ser mi hermana mayor e hija de nuestro padre que lo queremos mucho, aunque ya no esté con nosotros.

A mi familia que son lo más importante que puede existir en la vida de una persona, en especial a mi tía Maritza, mi prima Lianni y su marido Eduardo que a pesar de que los conocí hace menos de 2 años, me han apoyado y ayudado mucho, los quiero a todos. También a mi prima Soleika, que más que mi prima es como una tía para mí, pues siempre me ha ayudado en todo lo que ha podido, a su marido Armandito, a mis primos Michel y Michael, que más que eso, son mis hermanos, por poder contar con ellos para lo que sea, al igual que ellos pueden contar conmigo porque se lo merecen por todo su apoyo brindado, igual que a sus padres, mi tía Nora y Conrado. A mi prima Liudis por ser como es conmigo, por apoyarme y quererme tanto, por compartir tantos momentos de alegría y tristeza, igual que mi tía Marta, su mamá. A mi primo Alexis, que ese si me ayuda en todo lo que puede, otro hermano más que tengo. A mi prima Elizabeth que desde q llegué aquí a la uci siempre ha estado pendiente de mí.

Quiero agradecerle a mi familia en general, a todos los que me han ayudado en algún momento de mi vida, pues mi familia es muy grande y no me alcanzarían las hojas para mencionarlos a todos porque son muchos los que me han ayudado de una forma u otra, a todos muchas gracias...

A Arnaldo, que ya lo conocía porque es de mi barrio, mi amigo, que me ayudó y me encaminó cuando llegué aquí y no conocía a nadie, aunque aquí se hacen amistades muy rápido, que me prestaba su laptop para estudiar ip y el mismo me repasaba, aunque no me fue muy bien con el ip.

A mis amigas Dayani, Anabel, Margarita por compartir tantos momentos juntos de alegría, de fiestas, si porque no sé si en algún momento me vieron triste porque no me caracterizo por eso, por estudiar juntos y poder aprobar todo en las finales, que sin ellas todo hubiera sido más difícil para mí.

A Liz Yanet, que logrado tocarme el corazón por su forma de pensar y de ser conmigo, por compartir tantos momentos buenos y malos, por lograr que yo la pueda hacer reír cuando está triste, aunque no es tan difícil para mí hacer reír, es algo de mi naturaleza.

A Liosvel, que cada vez que necesitaba ayuda en algo, nunca me decía que no, siempre tenía tiempo para ayudar a sus amigos.

A mis amigos de la facultad 1, los amigos de fiestas, de pasarla bien, esos son el Javi, Humberto, Elvis, Iván, el Rafa.

A Yosvani y Yanetsy, mis amigos, compañeros que compartimos muchos momentos, que me apoyaron siempre que podían, que me daban consejos y me regañaban por mis locuras. A mis antiguos compañeros de cuarto, Pi, Yasser, Randy, Damián

A mis tutores, por su empeño y apoyo para que pudiera formarme como un buen ingeniero, al igual que a todos los profesores que me enseñaron a lo largo de mi carrera.

A todos los de mi aula, desde primero hasta quinto, por ser buenos compañeros de estudio y de fiesta, por compartir tantos momentos juntos.

Ahora dejando para lo último lo mejor de lo mejor, mis amistades más antiguas desde que entré a la UCI, así para mencionar son unos cuantos, empezamos por Luisito, el primero que conocí, mi hermano, mi amigo, mi compañero de fiesta, el Flaco, que puedo decir, vaya todo lo bueno que se pueda decir, mi amigo, el que me ayudaba en las clases, el que me regañaba, vaya muchas cosas para contar, a Gustavo todo lo mismo mencionado anteriormente, lo que esta vez también lo regañaba yo a él, a Lenia y Lili que fueron unas de las primeras personas y amigas que conocí cuando llegué y compartimos buenos momentos. Al chiki, el lima, Yailin, Daylen, Shirley, Mercedes, Félix, Richard muchos momentos buenos compartidos. A Gloria, Yanelis y Jaime, que siempre me apoyaron y ayudaron, lo mismo para estudiar que dándome consejos, también las fiestas que compartíamos juntos, vaya muchas cosas.

A Luisito, ese pinareño, mi paisa, el primero en conocer al entrar a la escuela, mi hermano, mi amigo, muchos buenos momentos de locuras y alegrías compartidos.

Quiero agradecerles a todos en general, sin que nadie se ponga bravo, a todos los que me ayudaron y formaron parte de mi vida aquí en la UCI, muchas gracias.

A esta universidad por todos los años vividos aquí, por darme la oportunidad de hacerme ingeniero y de conocer a personas maravillosas, que, si volviera a nacer, pidiera la UCI de nuevo porque aquí he pasado los mejores años de mi vida.

A todos muchas gracias...

## Resumen

A menudo, cuando las empresas comienzan la búsqueda para adquirir un sistema que abra puertas y barreras o que registre la jornada diaria de sus empleados, no tienen muy claro que es un control de acceso o de presencia. Confundiéndolos muchas veces e incluso pensando que son lo mismo. El presente trabajo tiene como objetivo desarrollar un sistema que permita controlar el acceso a áreas restringidas y la entrada y salida a los locales en la Universidad de las Ciencias Informáticas (UCI) mediante la utilización de una placa de *hardware* libre, en este caso arduino, posibilitando el envío de datos hacia un servidor web mediante un *ethernet shield* y comprobar si una persona tiene acceso o no a un determinado local. Para el desarrollo del sistema se empleó la tecnología llamada identificación por radio frecuencia (*RFID*) en el envío y recibo de datos entre las identificaciones de las personas y el servidor web permitiendo controlar el acceso de individuos a diferentes locales. También se utilizó la metodología de desarrollo AUP-UCI, el lenguaje de programación PHP desde el lado del servidor y C++ del lado del cliente. Como resultado se logra una aplicación web capaz de gestionar personas y locales y de controlar el acceso a áreas restringidas y la entrada y salida a locales en la UCI.

**Palabras claves:** acceso, arduino, control, *RFID*, seguridad, sistema.

## **Abstract**

Often, when companies begin the search to acquire a system that opens doors and barriers or that registers the daily work of their employees, they are not very clear that it is an access or presence control. Confusing them many times and even thinking that they are the same. The present work has as objective to develop a system that allows to control the access to restricted areas and the entrance and exit to the premises in the University of the Informatics Sciences (UCI) by means of the use of a plate of free hardware, in this case arduino, making it possible to send data to a web server using an ethernet shield and check whether a person has access to a certain location or not. For the development of the system, the technology called radio frequency identification (RFID) was used in the sending and receiving of data between the identifications of the people and the web server allowing to control the access of individuals to different premises. The AUP-UCI development methodology was also used, the PHP programming language from the server side and C++ from the client side. As a result, a web application capable of managing people and premises and controlling access to restricted areas and entry and exit to premises in the UCI is achieved.

**Keywords:** access, arduino, control, RFID, security, system.

## Índice

Introducción .....	1
Capítulo 1: Fundamentación teórica y metodológica.....	5
Introducción.....	5
1.1 Conceptos asociados .....	5
1.2 Tipos de tecnologías para el control de acceso .....	6
1.2.1 Sistemas biométricos .....	6
1.2.2 Código de barras .....	7
1.2.3 Tarjetas magnéticas.....	7
1.2.4 Tecnología RFID.....	8
1.3 Ámbito internacional .....	10
1.3.1 Control de Activos con Tecnología RFID en la empresa Steelcase México.....	11
1.3.2 Control de materiales de trabajo en hospitales de Estados Unidos .....	11
1.3.3 Logística en H&M Bay.....	11
1.4 Ámbito nacional.....	12
1.4.1 Balitas automatizadas .....	12
1.4.2 Aplicación y mesa multitáctil, Palacio del Il Cabo en Cuba.....	13
1.5 Valoración sobre los tipos de sistemas de control de acceso .....	13
1.6 Metodología de desarrollo .....	13
1.7 Propuesta de herramientas y tecnologías a utilizar.....	14
1.7.1 Lenguaje de Modelado UML 2.1 .....	14
1.7.2 Herramienta CASE: Visual Paradigm for UML 5.0 .....	14
1.7.3 Módulo RC522:.....	15
1.7.4 Arduino UNO:.....	15
1.7.5 Ethernet Shield: .....	16
1.7.6 Marco de trabajo(Framework) .....	17
1.7.7 Servidor de aplicaciones web.....	18

1.8 Lenguaje de programación .....	18
1.9 Entorno Integrado de Desarrollo(IDE).....	19
1.10 Sistema gestor de base de datos .....	20
1.10.1 Tipos de sistemas gestores de bases de datos .....	20
1.10.2 Selección del sistema gestor de base de datos a utilizar.....	23
1.10.3 PhpMyAdmin 2.2.6 para el manejo de MySQL.....	23
1.11 Relación de costo.....	23
Conclusiones del capítulo .....	24
Capítulo 2: Análisis y diseño de la solución propuesta.....	25
Introducción.....	25
2.1 Propuesta de solución .....	25
2.1.1 Diagrama de flujo del sistema .....	27
2.2 Requisitos de Software .....	28
2.2.1 Requisitos Funcionales .....	28
2.2.2 Requisitos No Funcionales.....	29
2.3 Descripción de Historias de Usuario (HU) .....	30
2.4 Arquitectura del software .....	33
2.4.1 Descripción de la arquitectura del sistema .....	33
2.5 Patrones de diseño.....	35
2.5.1 Patrones GRASP .....	35
2.5.2 Patrones GOF.....	37
2.6 Modelo de datos.....	37
Conclusiones del capítulo.....	38
Capítulo 3: Implementación y pruebas .....	39
Introducción.....	39
3.1 Estándares de código.....	39
3.2 Modelo de implementación.....	39

3.2.1 Diagrama de componentes .....	40
3.3 Modelo de despliegue .....	40
3.3.1 Descripción del diagrama de despliegue.....	41
3.4 Pruebas de software.....	41
3.4.1 Pruebas de Aceptación .....	41
3.4.2 Pruebas de Caja Blanca .....	53
Conclusiones del capítulo.....	55
Conclusiones generales.....	56
Recomendaciones .....	57
Referencias Bibliográficas.....	58
Bibliografía.....	62
Anexos.....	63

## Índice de tabla

Tabla 1:	Comparación entre tecnologías de control de acceso .....	10
Tabla 2:	Modelos de arduino.....	24
Tabla 3:	Costo del prototipo a utilizar .....	24
Tabla 4:	Descripción de los roles del sistema .....	27
Tabla 5:	HU del RF1 .....	30
Tabla 6:	HU del RF2 .....	31
Tabla 7:	HU del RF6 .....	32
Tabla 8:	Caso de prueba aceptación 1.....	42
Tabla 9:	Caso de prueba aceptación 2.....	43
Tabla 10:	Caso de prueba aceptación 3.....	43
Tabla 11:	Caso de prueba aceptación 4.....	44
Tabla 12:	Caso de prueba aceptación 5.....	44
Tabla 13:	Caso de prueba aceptación 6.....	45
Tabla 14:	Caso de prueba aceptación 7.....	45
Tabla 15:	Caso de prueba aceptación 8.....	45
Tabla 16:	Caso de prueba aceptación 9.....	46
Tabla 17:	Caso de prueba aceptación 10.....	46
Tabla 18:	Caso de prueba aceptación 11.....	47
Tabla 19:	Caso de prueba aceptación 12.....	47
Tabla 20:	Caso de prueba aceptación 13.....	48
Tabla 21:	Caso de prueba aceptación 14.....	48
Tabla 22:	Caso de prueba aceptación 15.....	49
Tabla 23:	Caso de prueba aceptación 16.....	49
Tabla 24:	Caso de prueba aceptación 17.....	50
Tabla 25:	Caso de prueba aceptación 18.....	50
Tabla 26:	Caso de prueba aceptación 19.....	51

Tabla 27:	Caso de prueba aceptación 20 .....	51
Tabla 28:	Caso de prueba aceptación 21 .....	51
Tabla 29:	HU del RF3.....	63
Tabla 30:	HU del RF4.....	63
Tabla 31:	HU del RF5.....	64
Tabla 32:	HU del RF7.....	66
Tabla 33:	HU del RF8.....	66
Tabla 34:	HU del RF9.....	67
Tabla 35:	HU del RF10.....	68
Tabla 36:	HU del RF11.....	69
Tabla 37:	HU del RF12.....	69
Tabla 38:	HU del RF13.....	70
Tabla 39:	HU del RF14.....	70
Tabla 40:	HU del RF15.....	71
Tabla 41:	HU del RF16.....	71
Tabla 42:	HU del RF17.....	72
Tabla 43:	HU del RF18.....	73
Tabla 44:	HU del RF19.....	73
Tabla 45:	HU del RF20.....	74
Tabla 46:	HU del RF21.....	75

## Índice de figuras

Figura 1: Cerraduras <i>RFID</i> .....	25
Figura 2: Diagrama de flujo de la propuesta de solución .....	28
Figura 3: Patrón de diseño MVC .....	35
Figura 4: Modelo Entidad-Relación .....	38
Figura 5: Diagrama de componentes gestionar persona .....	40
Figura 6: Diagrama de despliegue.....	41
Figura 7: Resultados de las pruebas de caja negra.....	53
Figura 8: Resultados de las pruebas de unidad.....	55
Figura 9: Diagrama de componentes gestionar tarjeta .....	76
Figura 10: Diagrama de componentes gestionar local .....	76
Figura 11: Diagrama de componentes obtener datos de persona dado usuario .....	77
Figura 12: Diagrama de componentes acceder a local, notificar y comprobar acceso a local .....	77

## Introducción

En la actualidad la sociedad está viviendo una época de grandes avances tecnológicos. Uno de los elementos más relevantes de este fenómeno tiene que ver con su carácter global, pues el progreso tecnológico no se limita a países del primer mundo, sino que se extiende a países emergentes.

En este marco se han podido identificar experiencias innovadoras y relevantes que suponen la utilización de las Tecnologías de la Información y las Comunicaciones (TIC) para la resolución de problemas o necesidades sociales de poblaciones de bajos recursos. Estas incluyen la electrónica como la tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual (1).

A lo largo de todo este desarrollo tecnológico que está viviendo la sociedad, la búsqueda de un eficiente manejo de la información ha sido objetivo primordial. Teniendo en cuenta que la forma de acceder a la información, servicios y aplicaciones, se extiende cada vez más y también aumentan los riesgos de amenazas para los usuarios; surge la necesidad de crear sistemas de control de acceso que brinden técnicas seguras para limitar el libre acceso del público en general a la información, lugar o recurso que deban ser protegidos.

El control de acceso es la parte principal de la pirámide de criticidad de sistemas de seguridad. Así que la elección de una solución fiable es crítica. El uso de tecnologías para ello ha crecido enormemente en los últimos años. Sin embargo, existen en el mercado muchas soluciones que no cumplen con los requisitos mínimos de seguridad, confiabilidad y estabilidad que estos sistemas requieren (2).

Con el control de acceso se puede restringir fácilmente días y horarios de acceso y emitir informes detallados de la actividad del usuario. Una sola tarjeta puede abrir todas las puertas desde que, obviamente, tenga los permisos para hacerlo (2).

Inicialmente para garantizar la seguridad se recurría a las alarmas antirrobo, las cuales tenían una función única: emitir un sonido ante la detección de una persona o intruso; estas alarmas sonoras no fueron lo suficientemente efectivas para poder asegurar la tranquilidad de las viviendas o locales (4). Esto se debe a que dichas alarmas tenían una alta probabilidad para que su mecanismo fallara o que simplemente el sensor se activara por su sensibilidad en cualquier momento o circunstancia no deseada.

En sus comienzos los sistemas de control de acceso usaron con frecuencia como tecnología los teclados por Número de Identificación Personal (*PIN*) llamados así por sus siglas en inglés, los cuales fueron reemplazados lentamente por sistemas con tarjetas magnéticas y código de barra. Con los avances tecnológicos se hizo presente la tarjeta de proximidad y al mismo tiempo aparecieron los lectores

biométricos (5). Los usos de estas tecnologías brindan la posibilidad de aumentar el nivel de seguridad en las instalaciones militares, centros de investigación, centros escolares y demás instituciones que así lo requieran.

Una de las tecnologías que está cobrando auge a nivel mundial es la identificación por radiofrecuencia (*RFID* por sus siglas en inglés), esta no es más que un sistema para la comunicación inalámbrica entre dos o más objetos, donde uno emite señales de radio y el otro responde en función de la señal recibida; la comunicación se realiza a través de una antena con un *transponder* (también conocido como *tag* en inglés o etiqueta). Asimismo esta es útil donde tengan que realizarse continuos registros de datos, sin contacto y sin la necesidad de campo visual, ya sea para identificar, rastrear y gestionar productos, documentos, objetos, personas, animales (6).

Un sistema de control de acceso *RFID* provee una solución sencilla y eficiente, esto se debe a que las Identificaciones *RFID* pueden ser leídas a distancias mucho mayores en comparación con medios tradicionales y la información contenida en cada credencial puede ser sobre escrita repetidamente. Además, los aumentos de las distancias de lectura permiten habilitar otras tecnologías como el activar cámaras de vigilancia cuando un empleado entra a una determinada área. Más aun, múltiples credenciales pueden ser leídas al mismo tiempo. También la información sobre horas de entrada, asistencia y labores realizadas puede ser fácilmente monitoreada y almacenada en una base de datos (7).

Las organizaciones hoy en día requieren un sistema de control de acceso funcional y eficiente debido a que deben mejorar la seguridad para limitar el acceso a áreas restringidas, monitorear la entrada y salida de los empleados y realizar mejoras para prevenir pérdidas en los locales de trabajo.

Actualmente, existen grandes deficiencias en los sistemas tradicionales de control de acceso basados en la identificación por tarjetas, tales como los códigos de barra y cintas magnéticas; estas tecnologías dependen del contacto directo con el dispositivo lector o bien de colocar las identificaciones cerca del lector, generando con el tiempo desgaste físico de las tarjetas. Además, los códigos de barra solamente pueden ser utilizados una vez y la información contenida en ellos no puede ser actualizada, y las cintas magnéticas al rayarse o ser expuestas a campos magnéticos pierden la información que contienen (8). Estas limitaciones resultan engorrosas para mantener un control de acceso eficiente y además provocan pérdidas de tiempo.

Cuba no está exenta de los problemas mencionados anteriormente, en la actualidad hay locales que tienen un control de acceso con tecnología muy obsoleta o se basan en el contrato de personal para regular dicho acceso, lo cual provoca deficiencias en la seguridad de las áreas; además de que se utilizan mecanismos que generan retrasos y control insuficiente en el monitoreo del cumplimiento de la jornada laboral.

En la Universidad de las Ciencias Informáticas(UCI) se realiza un control de acceso mediante los códigos de barra en los comedores y en las entradas a la universidad; esta tecnología mencionada anteriormente requiere una verificación visual directa. Además de que los lectores no pueden leer etiquetas que estén dañadas, sucias o manchadas pues provoca como consecuencia un control insuficiente en los lugares donde se utiliza dicha tecnología.

A partir de la situación problemática anteriormente expuesta, se plantea como **problema de la investigación**: ¿Cómo controlar el acceso a áreas restringidas y la entrada y salida a los locales en la UCI?, teniendo como **objeto de estudio**: Sistemas de control de acceso. Enmarcado en el **campo de acción**: Sistemas de control de acceso basado en *RFID*.

Se propone como **objetivo general**: Desarrollar un sistema que permita controlar el acceso a áreas restringidas y la entrada y salida a los locales en la UCI.

Para dar solución al objetivo planteado se proponen como tareas de la investigación:

1. Estudio del estado del arte del tema en cuestión valorando las funcionalidades de sistemas de control de acceso existentes.
2. Identificación de los requisitos funcionales para el desarrollo de la solución propuesta.
3. Generación de los artefactos relacionados con el análisis y diseño de la solución propuesta.
4. Propuesta de la arquitectura del sistema de control de acceso cumpliendo con las políticas de *software* libre.
5. Propuesta de roles en el sistema para la obtención de la información generada en el mismo de acuerdo a los permisos del cliente.
6. Diseño e implementación de la solución propuesta.
7. Validación del sistema a través de la realización de pruebas.

Con el objetivo de desarrollar las tareas mencionadas anteriormente se trabaja con los siguientes **métodos de la investigación**:

#### **Métodos teóricos de investigación**

- Histórico-Lógico: Es utilizado para analizar la evolución histórica de soluciones similares en los sistemas de control de acceso utilizados en el mundo, las tendencias actuales y su utilización en Cuba para determinar si resuelven el problema planteado.

- **Analítico-Sintético:** Para el análisis de la documentación existente relacionada con el tema, extraer los elementos más importantes y necesarios, de manera que permita sintetizar todo lo obtenido. Se utilizará, además, para seleccionar las herramientas y tecnologías a utilizar durante el desarrollo de la herramienta.
- **Modelación:** Es utilizado para modelar los procesos del negocio en cuestión y los elementos necesarios para la implementación. Se empleará para realizar los diagramas necesarios para darle cumplimiento a los requisitos funcionales y no funcionales asociados al sistema.

### **Métodos Empíricos de Investigación**

- **Observación:** Se utiliza para obtener conocimiento acerca del comportamiento del objeto de investigación tal y como este se da en la realidad, es una manera de acceder a la información directa e inmediata sobre el proceso, fenómeno u objeto que está siendo investigado. Se empleará para realizar el estudio del estado del arte acerca de los diferentes sistemas de control de acceso existentes.

El presente documento consta de tres capítulos que a su vez se dividen en epígrafes y estos en sub-epígrafes de acuerdo al nivel de detalle que requiere el contenido abordado en cada uno de ellos. A continuación, se explica brevemente el contenido de cada capítulo:

**Capítulo I. Fundamentación teórica y metodológica:** En este capítulo se precisará la base teórica de la presente investigación, incluye el análisis de la información existente acerca del tema a tratar y las tendencias actuales que existen en el mundo. También contendrá una descripción dando un sustento teórico a la selección de las herramientas y *frameworks* a utilizar para resolver el problema planteado.

**Capítulo II. Análisis y diseño de la solución propuesta:** En este capítulo se realizará una breve descripción de la solución propuesta, sus requisitos funcionales y no funcionales. Asimismo, se representarán los artefactos generados a través del proceso de desarrollo del sistema.

**Capítulo III. Implementación y pruebas:** En este capítulo se mostrará parte de la implementación de la solución, especificando el estándar de codificación. Contendrá además las pruebas que le serán aplicadas al sistema a desarrollar y el análisis de los resultados obtenidos en este proceso.

# Capítulo 1: Fundamentación teórica y metodológica.

## Introducción

En el presente capítulo se realizará un estudio de los conceptos asociados al control de acceso y un análisis sobre los principales sistemas de control de acceso existentes, basado principalmente en la identificación por radiofrecuencia. Además, se hará un estudio de las herramientas y tecnologías, posteriormente se argumenta la selección de las que se emplearán en el desarrollo e implementación del sistema. Se justificará la utilización de la metodología de desarrollo de software, lenguajes de programación, herramienta CASE y lenguaje de modelado.

### 1.1 Conceptos asociados

A continuación, se mostrarán algunos conceptos tratados durante el transcurso de la investigación, a fin de introducir el tema y lograr una mejor comprensión del trabajo en cuestión.

**Control:** es una de las principales actividades administrativas dentro de las organizaciones. También es el proceso de verificar el desempeño de distintas áreas o funciones de una organización. Usualmente implica una comparación entre un rendimiento esperado y un rendimiento observado, para verificar si se están cumpliendo los objetivos de forma eficiente y eficaz y tomar acciones correctivas cuando sea necesario (9).

El **control de acceso** es una técnica de seguridad que se puede usar para regular quien o que puede ver o usar recursos en un entorno informático (10).

Un **control de acceso** tiene por objeto impedir el libre acceso del público en general a diversas áreas que se denominan protegidas donde solo puede haber personal técnicamente capacitado. Deberán definirse los permisos, reglas o privilegios de cada uno de los que podrán acceder a determinada zona protegida. Estos privilegios podrán depender de la categoría o rango de la persona dentro de la empresa, de su función, de un determinado horario en el que puede ingresar o salir (11).

Después de un análisis de los conceptos antes mencionados se observó que todos tienen puntos en común como son: restringir o permitir el libre acceso de los usuarios a la información, recursos o áreas protegidas en un horario determinado. Lo que permitió elaborar el siguiente concepto:

**Control de acceso:** es la acción de limitar adquirir información, recursos o la entrada a lugares que se consideren protegidos, dependiendo de la identidad de un usuario.

Un **sistema de control de acceso** hace referencia al mecanismo que en función de la identificación ya autenticada permite acceder a datos o recursos. Básicamente se encuentran sistemas de controles de acceso en múltiples formas y para diversas aplicaciones como lo son peatonal, vehicular y de personal.

## **1.2 Tipos de tecnologías para el control de acceso**

Como elemento importante de esta investigación, se realizará un estudio y caracterización de algunas tecnologías informáticas que proporcionan el control de acceso a áreas restringidas y la entrada y salida a locales. Dentro de las diferentes tecnologías utilizadas se encuentran el código de barras, las tarjetas magnéticas, sistemas biométricos y la tecnología de identificación por radiofrecuencia.

### **1.2.1 Sistemas biométricos**

La biometría informática permite restringir el acceso a locales, centros laborales y establecimientos. Mediante esta tecnología se puede realizar la identificación y reconocimiento de personas, permitiendo elegir cuales tienen autorización para acceder a ciertos lugares. Pero la biometría informática no solo se aplica en controlar el acceso de individuos, sino que también aporta un gran apoyo en el campo forense y la vigilancia (12).

Los sistemas biométricos fundamentan sus decisiones de reconocimiento mediante una característica personal, donde los lectores reconocen automáticamente la característica física de un sujeto eliminando por completo el uso de tarjetas electrónicas o magnéticas. Las principales características físicas que se trabajan en el reconocimiento de los mismos son: reconocimiento de iris, reflexión de retina, geometría de la mano, geometría facial, termografía mano-facial, huellas dactilares y patrón de voz (13).

La principal ventaja de la biometría es que es más eficiente y segura que los sistemas tradicionales como las contraseñas, llaves, o tarjetas pues los rasgos físicos son únicos de uno mismo y no un elemento externo, o sea, ni se pierden ni se olvidan (12).

Las desventajas de esta tecnología son su alto costo de implementación (por los lectores que se manejan para detectar los rasgos físicos), la reducida velocidad de lectura (comparada con la de otros sistemas como los que usan *RFID*) y la carencia de una eficiencia necesaria para grandes corporaciones pues los retardos en las lecturas de personal disminuirían tiempos en las labores (13).

En relación con lo antes mencionado se puede definir que un **sistema biométrico** es un medio computarizado que permite la identificación de individuos utilizando rasgos físicos o de conducta, es decir, verificar su identidad.

### **1.2.2 Código de barras**

El código de barras constituye una forma de entrada de datos; es un arreglo en paralelo de barras y espacios que contiene información codificada. Esta información puede ser leída por dispositivos ópticos, los cuales envían la información leída hacia una computadora tal como si se hubiera tecleado (14).

El código de barras es conocido como la tecnología de captura automática de información que permite identificar productos y servicios mediante un código numérico que por lo general se combina con uno alfabético, es un sistema sencillo para ser implementado en cualquier tipo de organización, independientemente de su tamaño o función (15).

La principal ventaja de dicha tecnología es que su implementación es muy barata pues la creación de códigos no es muy compleja y es de fácil aplicación a las tarjetas que contendrán los códigos. Además, los equipos de lectura e impresión son flexibles y fáciles de conectar e instalar (13).

Las desventajas de esta tecnología son que los lectores de códigos de barras no pueden leer etiquetas que estén arrugadas, sucias o manchadas y los lectores ópticos de este sistema requieren una verificación visual directa(13).

En consideración con lo antes mencionado se puede definir que **los códigos de barra** son una técnica de codificación gráfica que representa datos en forma de barras y espacios de diferentes dimensiones y representaciones que ha ayudado a los comerciantes en la identificación de productos y precios.

### **1.2.3 Tarjetas magnéticas**

Las tarjetas magnéticas son medios que se basan en la lectura de una banda magnética. Utilizan señales electromagnéticas para registrar y codificar información en una franja que puede ser leída por una máquina para identificación instantánea (16).

Las tarjetas magnéticas contienen una banda magnética que posee un código que permite identificarse rápidamente. Este sistema utiliza señales electromagnéticas para registrar y codificar la información. Una de las aplicaciones más comunes de esta tecnología son las tarjetas de crédito (13).

Esta tecnología presenta como ventajas altos niveles de seguridad en cuanto a que contiene una clave privada o pin, esta presenta un cifrado de información y proporciona agilidad en el acceso, permite dar identificación única al poseedor (13).

Como desventajas, su uso continuo las deteriora físicamente debido a la fricción en el momento de la lectura; también si la tarjeta es acercada a una fuente electromagnética, relativamente fuerte, la información contenida en ella puede ser modificada, con lo cual pierde su utilidad (13).

De acuerdo con lo mencionado anteriormente las **tarjetas magnéticas** son tarjetas a las que se les ha incorporado una banda magnética en el proceso de fabricación. Se lee dicha franja por contacto físico o pasándolo por una cabecera lector. También pueden ser magnetizadas de nuevo en caso de desmagnetizarse por la presencia de un campo magnético.

#### **1.2.4 Tecnología RFID**

*RFID* es una tecnología para la identificación de objetos, personas y animales a distancias sin necesidad de contacto o línea de vista; se trata de una técnica aplicable en situaciones muy variadas, que sirve de apoyo a un conjunto muy extenso de aplicaciones en una gran variedad de ámbitos, desde la trazabilidad y control de inventario, hasta la localización y seguimiento de personas y bienes, o la seguridad en el control de accesos a establecimientos comerciales, militares, laborales y educativos (13).

La tecnología *RFID* ha adoptado la industria mediana y grande de los países con comercio internacional, con el fin de obtener ventajas competitivas. Esta utiliza las ondas de radio para enviar y recibir datos en tiempo real sin necesidad de estar conectado a una red alámbrica (17).

Al profundizar sobre lo mencionado anteriormente se puede constatar que la **tecnología RFID** es un conjunto de técnicas para la identificación de objetos a distancia sin necesidad de contacto, ni siquiera visual, esta no es más que un sistema para comunicarse de forma inalámbrica entre dos o más objetos, donde uno emite señales de radio y el otro responde en función de la señal recibida. Estos datos remotos se almacenan en tarjetas, etiquetas o *tags RFID* mediante un microchip que está incorporado a una antena.

Las etiquetas *RFID*, son unos dispositivos de pequeño tamaño fácil de adherir a cualquier objeto. Estos *tags*, tienen en su interior antenas de radiofrecuencia las cuales les permiten responder a peticiones de un emisor-receptor. Hay 3 tipos, las pasivas, las activas y las semi-pasivas. Las pasivas no necesitan alimentación, mientras que las activas sí y las semi-pasivas son una combinación entre las activas y las pasivas.

Las ventajas de la tecnología *RFID* son las siguientes **(13)**:

Flexibilidad de Información: con respecto a la información dinámica, las etiquetas *RFID* son capaces de realizar operaciones de lectura y escritura, permitiendo la actualización de información en tiempo real.

Durabilidad: para mayor protección, las etiquetas *RFID* pueden ser insertadas en sustratos de plástico duro u otros materiales. La naturaleza de las etiquetas *RFID* les permite perdurar más que las de código de barras y tarjetas magnéticas.

Seguridad: por el diseño tecnológico que tienen las tarjetas no pueden duplicarse, cada *tag* posee un código único, por lo cual no permite que varios usuarios puedan tener una tarjeta duplicada.

Inventarios de alta velocidad: múltiples dispositivos pueden ser leídos simultáneamente, lo que ahorra un tiempo significativo en comparación con otras tecnologías a las cuales los productos deben ser pasados uno por uno.

Reescribible: algunos tipos de *tag RFID* pueden ser leídas y escritas en múltiples ocasiones.

Para una mayor comprensión sobre las ventajas y desventajas que existen entre las diferentes tecnologías usadas para el control de acceso se muestra la siguiente tabla (13):

Tabla 1: Comparación entre tecnologías de control de acceso

<b>Tecnología de lectura</b>	<b>Seguridad</b>	<b>Desgaste de tarjeta</b>	<b>Desgaste de lector</b>	<b>Costo de mantenimiento</b>	<b>Precio tarjeta</b>	<b>Precio lector</b>
<u>Código de barras</u>	Baja	Medio	Bajo	Medio	Bajo	Medio
<u>Tarjeta magnética</u>	Medio	Alto	Muy alto	Alto	Muy bajo	Bajo
<u>Sistema biométrico</u>	Muy alta	No posee	Bajo	Medio-Alto	No posee	Muy alto
<u>Tecnología <i>RFID</i></u>	Alta	No posee	No posee	Muy Bajo	Medio-Bajo	Medio

Después de haber analizado dicha tabla se puede arribar a la conclusión de que la tecnología *RFID* es la más eficiente para implementar un sistema de control de acceso pues la misma presenta una seguridad de uso alta debido a que las tarjetas contienen un número de identificación único que no puede ser modificado, no posee desgaste de tarjeta ni de lector en el transcurso del tiempo en su utilización con respecto a las tecnologías restantes porque no se necesita contacto directo entre el dispositivo que emite la señal de radio y el que la recibe. También el costo de mantenimiento es muy bajo, al igual que el precio de las tarjetas y del lector es medio y bajo, siendo estos, aspectos a tener en cuenta para la implementación de dicho sistema.

### 1.3 **Ámbito internacional**

Como elemento importante de esta investigación, se realizará un estudio y caracterización de algunos sistemas informáticos existentes a nivel internacional que usan la tecnología *RFID* en diferentes empresas o instituciones.

### **1.3.1 Control de Activos con Tecnología RFID en la empresa Steelcase México**

Los bancos y otras empresas de la ciudad de México están utilizando etiquetas pasivas para hacer un seguimiento de su mobiliario y están probando un sistema que usa etiquetas activas para detectar cuando se están utilizando áreas específicas de las oficinas.

El sistema de gestión de activos está compuesto por etiquetas pasivas que se colocan en la parte inferior de las sillas y que transmiten un número de identificación único a un dispositivo portátil o a un lector fijo instalado como portal dentro de las oficinas del usuario. También incluye una aplicación de *software* que los usuarios pueden instalar para vincular los números de identificación de las etiquetas a los detalles relacionados con los muebles de oficina, incluida no solo la descripción de un artículo, sino también a que lugar pertenece dicho activo y a quien está asignado (18).

El control de activos utilizado por Steelcase México permite que estos se conviertan en una de las mejores inversiones de la empresa, impactando en los siguientes aspectos:

- Financiero: debido a que se debe reflejar en las cuentas bancarias del estado el valor de las amortizaciones, ingresos o gastos que estos generen.
- Fiscal: permite generar una oportunidad de optimización fiscal, pues ayuda a tomar los criterios más convenientes según las normativas impuestas.
- Costos: permite reducir el valor de inversiones (terrenos, edificios, maquinarias, entre otros) haciendo cargos que afecten al estado de resultados a través del tiempo.

### **1.3.2 Control de materiales de trabajo en hospitales de Estados Unidos**

La empresa *McKesson* en Estados Unidos utiliza la tecnología *RFID* para el control en los hospitales de materiales de trabajo como marcapasos, implantes, entre otros. Cada herramienta tiene un *tag* integrado, permitiéndole al sistema tener un registro en tiempo real de todo el inventario que se tenga de implementos que sean utilizados por 37 enfermeras y médicos, este sistema admite el control de dichos instrumentos evitando diferentes robos que se puedan presentar al interior de la organización (19).

### **1.3.3 Logística en H&M Bay**

Esta compañía la cual se dedica al transporte de carga, logró mejorar su logística para el transporte de alimentos congelados. Este tipo de alimentos requiere de un cuidado primordial en la medida en que es necesario mantener la temperatura con el afán de evitar la descomposición de los diferentes artículos. H&M Bay incluyó un *hardware RFID* y una solución de *software* para localizar y registrar la información de plataforma en tiempo real, ofreciendo a los operadores una lectura instantánea del inventario a medida que entra, se mueve y sale de la instalación de distribución (20).

La forma de cómo mejoraron los procesos de esta compañía, fue por medio de la tecnología *RFID*, implementando este sistema en el centro de distribución de la empresa, lo cual generó eficiencia y productividad en el manejo de la carga para cumplir con los estándares de calidad requeridos por el gobierno y las compañías que fabrican este tipo de alimentos.

## **1.4 Ámbito nacional**

Como elemento importante de esta investigación, se realizará un estudio y caracterización de algunos sistemas informáticos existentes en el ámbito nacional que usan la tecnología *RFID* en diferentes empresas o instituciones.

### **1.4.1 Balitas automatizadas**

Cuba desarrolla un novedoso sistema informático para mejorar el control y la productividad con los cilindros de gas licuado, las popularmente conocidas como “balitas”. Este sistema ha sido desarrollado por la empresa Tecnomática, adscrita a la Unión Cuba-Petróleo (Cupet), y fue presentado como parte del panorama de las Tecnologías de la Información y las Comunicaciones en Cuba en la 16ª Convención y Feria Informática 2016.

El sistema ingenieril cuenta con una serie de componentes que abarcan todo el proceso productivo de las balitas. Estas balitas necesitan un seguimiento que brinde su ubicación y rinda cuentas de su estado de mantenimiento. Tiene sentido, sobre todo cuando se sabe que esta es una herramienta de uso doméstico que debe ser manipulada con mucha precaución.

El nuevo sistema elimina también el vale de llenado a mano por los operadores, con números como la fecha de recambio, que a veces son difíciles de entender, y en cambio imprime un comprobante semejante al de las tiendas recaudadoras de divisas, siendo esto una gran ventaja para su uso (21).

Otro aspecto interesante es cómo influye en los rendimientos dentro de la cadena productiva. Más allá de los puntos de venta, en las plantas de llenado existe una persona que comprueba el estado técnico de los cilindros de gas. Si alguno de los cilindros que van a ser rellenos presenta algún problema, el sistema lo expulsa de la cadena y lo envía a mantenimiento.

#### **1.4.2 Aplicación y mesa multitáctil, Palacio del II Cabo en Cuba**

Virtualware<sup>1</sup> ha desarrollado para Tecnalía<sup>2</sup> una aplicación turística interactiva instalada en 3 mesas multitáctiles en la Sala del Gabinete del Palacio del II Cabo: una adaptada para niños y otras dos para visitantes adultos. La aplicación *multitouch* permite el uso simultáneo en 4 zonas de trabajo mediante unos cubos que incorporan tecnología *RFID*, cada uno con una temática del centro. Cuando el visitante apoya sobre la pantalla de mesa moviliza el cubo activador y puede disfrutar en su visita de contenidos multimedia enriquecidos: imágenes, vídeos y un juego de bloques (22).

La aplicación de las mesas se alimenta de una base de datos en la nube, que mediante un gestor de contenidos remoto permite al cliente, modificar, eliminar y añadir información.

#### **1.5 Valoración sobre los tipos de sistemas de control de acceso**

Como resultado del estudio de las tecnologías utilizadas para el control de acceso a áreas restringidas, ninguno de estos sistemas es idóneos para desplegarlos en la UCI, teniendo en cuenta que estos fueron creados para instituciones con un objetivo específico y son sistemas propietarios que tiene un alto costo monetario; aunque sirven de apoyo para el desarrollo del sistema a implementar pues estos utilizan etiquetas pasivas, las cuales no necesitan alimentación eléctrica y son menos costosas que las activas que sí necesitan suministro de energía. También los registros de acceso de estos sistemas se realizan en tiempo real, lo cual permite un control más eficiente y detallado. Además, se tiene que implementar un nuevo sistema, capaz de dar solución a los problemas existentes en la universidad como lo son: el control insuficiente a áreas restringidas y el mal monitoreo del cumplimiento de la jornada laboral, así como la entrada y salida a los locales en la UCI, este *software* a desarrollar será mediante identificación por radiofrecuencia teniendo en cuenta las ventajas que esta tecnología presenta mencionadas anteriormente.

#### **1.6 Metodología de desarrollo**

La metodología de desarrollo es un marco de trabajo que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los objetivos. Es el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y que hacer para realizarlos y finalizar una tarea. Estas se dividen en dos enfoques o ramas, las cuales son conocidas como metodologías ágiles y metodologías tradicionales. Las tradicionales centran su atención en llevar una documentación exhaustiva de todo el proyecto, la

---

<sup>1</sup> Virtualware es una empresa tecnológica. especializada en el desarrollo de productos y soluciones a medida basadas en tecnologías inmersivas e interactivas.

<sup>2</sup> Corporación Tecnalía es una Corporación Tecnológica con el principal objetivo de contribuir al desarrollo del entorno económico y social a través del uso y fomento de la Innovación Tecnológica.

planificación y control del mismo, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto en la fase inicial del desarrollo del proyecto. Por otro lado, las ágiles son convenientes para guiar proyectos de escaso volumen comercial que demanden una rápida implementación (23).

Proceso unificado ágil ajustada a la Universidad de las Ciencias Informáticas (AUP-UCI) es una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Esta tiene entre sus objetivos aumentar la calidad del *software* que se produce, de ahí la importancia de aplicar buenas prácticas, para ello es necesario apoyarse en el Modelo CMMI-DEV v1.3. El cual constituye una guía para aplicar las mejores habilidades en una entidad desarrolladora. Estas facilidades se centran en el desarrollo de productos y servicios de calidad (3).

Para el caso del desarrollo de la investigación, se utilizará la metodología AUP-UCI en su escenario No 4: Historias de Usuario(HU), para este escenario en específico no se modela el negocio, sólo se puede modelar el sistema a partir de las HU. Es aplicado a los proyectos que hayan evaluado el ejercicio a informatizar y como resultado obtengan un negocio muy bien definido. Además, el cliente siempre acompañará al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

## **1.7 Propuesta de herramientas y tecnologías a utilizar**

### **1.7.1 Lenguaje de Modelado UML 2.1**

Es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de *software* orientado a objetos (OO). Para realizar los modelos del sistema propuesto se hará uso del Lenguaje Unificado de Modelado 2.1 (*UML*, por sus siglas en inglés), porque tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue (24).

### **1.7.2 Herramienta CASE: Visual Paradigm for UML 5.0**

Las herramientas de ingeniería de *software* asistida por computadora (*CASE* por sus siglas en inglés), son aplicaciones computacionales en conjunto que soportan y ayudan al proceso de análisis y desarrollo de *software*. Las cuales sirven a los analistas de sistemas, ingenieros y desarrolladores, durante todos los pasos del ciclo de vida del desarrollo del *software* (25).

Para apoyar el modelado de los diagramas se hará uso de la herramienta *CASE Visual Paradigm for UML*, la cual soporta hasta la versión 2.1 de *UML* y permite modelar los procesos del negocio, la base de datos y

las clases del sistema de manera visual. Está diseñada para soportar el ciclo de vida completo del proceso de desarrollo del software, a través de la representación de todo tipo de diagramas. Con capacidad de ingeniería directa e inversa enfocada al negocio generando un *software* de mayor calidad.

### **1.7.3 Módulo RC522:**

RC522 utiliza un sistema avanzado de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56Mhz. Como se hará una lectura y escritura de la tarjeta, es necesario conocer las características de los bloques de memoria de la misma: la etiqueta que viene con el módulo *RFID*, cuenta con 64 elementos (0-63) donde se hace lectura o escritura. Cada bloque de memoria tiene la capacidad de almacenar hasta 16 Bytes. El número de serie consiste de 5 valores hexadecimales, se podría utilizar esto para hacer una operación dependiendo del número de serie (26).

Características:(27)

- Placa basada en chip MFRC522.
- Frecuencia de funcionamiento: 13.56MHz.
- Voltaje de suministro: 3.3V.
- Rango de lectura: Aproximadamente 3 cm con tarjeta y llavero suministrados.
- Interfaz *SPI*<sup>3</sup>.
- Velocidad máxima de transferencia de datos: 10Mbit / s.
- Dimensiones: 60 mm x 39 mm.

Este módulo de lector de *RFID* basado en RC522 fue seleccionado pues es de bajo costo, fácil de usar y se puede utilizar en una amplia gama de aplicaciones, es un módulo de lectura y escritura altamente integrado para comunicación sin contacto a una frecuencia de 13.56 MHz. Este dispositivo lector no permite leer varias identificaciones a la vez siendo esta una desventaja.

### **1.7.4 Arduino UNO:**

Es una tarjeta de desarrollo basada en el microcontrolador ATmega328. Tiene 14 entradas y salidas digitales, estas son signos que representan ciertos valores discretos que contienen información codificada y 6 entradas analógicas, las cuales son aquellas en la que los valores de la tensión o voltaje varían constantemente y pueden tomar cualquier valor. También corre con un resonador a 16 Mhz, tiene conexión USB, un Jack (conector análogo) para alimentación externa y un botón de *reset*. Esta tarjeta contiene todo

---

<sup>3</sup> SPI (del inglés *Serial Peripheral Interface*) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

lo que se necesita para utilizar el microcontrolador AVR ATmega; solo basta con conectarla al puerto USB de la computadora para comenzar a desarrollar algún proyecto (28).

Desde el punto de vista conceptual arduino es la unión de varios elementos de *hardware* y *software*. El *hardware* de este incluye a los siguientes elementos: microcontrolador, pines de conexión, fuente de alimentación y puerto de comunicación serial USB. Su lenguaje de programación está basado en C++, aunque utiliza C y soporta todas sus funciones del estándar de dicho lenguaje (29).

Teniendo en cuenta lo mencionado anteriormente se puede decir que arduino es una plataforma de prototipos electrónica de código abierto basada en una sencilla placa con entradas y salidas (E/S), analógicas y digitales. Se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios.

Esta herramienta fue seleccionada pues le permite a cualquier persona poder crear y diseñar un prototipo de una forma muy fácil, incluso sin requerir algún conocimiento de electrónica debido a que existe una amplia documentación y ejemplos sencillos desarrollados con dicha herramienta.

#### **1.7.5 Ethernet Shield:**

El *ethernet shield* da la capacidad de conectar un arduino a una red cableada. Es la parte física que implementa la pila de protocolos TCP/IP, está basada en el chip Wiznet W5100. También provee un conector estándar RJ45. Este dispone de unos conectores que permiten conectar a su vez otras placas encima y juntar sobre la placa arduino.

Arduino usa los pines digitales 10, 11, 12 y 13 (SPI) para comunicarse con el W5100 en la *ethernet shield*. Estos pines no pueden ser usados para entradas y salidas genéricas. Entre sus principales características se encuentran las siguientes (30):

Características:

- Opera a 5V suministrados desde la placa de arduino.
- El controlador es el W5100 con 16K de buffer interno, no consume memoria.
- Se comunica con el microcontrolador por el bus *SPI*, por lo tanto, para usarlo siempre se debe incluir la librería *SPI.h*.
- Soporta hasta 4 conexiones simultáneas.
- Usa la librería *Ethernet.h* para manejar el *shield*.

- Dispone de un lector de tarjetas micro-SD <sup>4</sup> que puede ser usado para guardar ficheros y servirlos sobre la red, para ello es necesaria la librería *SD*.

Esta herramienta fue seleccionada pues permite la comunicación por red, como por ejemplo enviar datos a un servidor local o incluso comunicarse con la placa mediante Internet.

### **1.7.6 Marco de trabajo(Framework)**

**Symfony2.8:** es un proyecto PHP de *software* libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional. Entre sus principales características están las siguientes(31):

- Su código y el de todos los componentes y librerías que incluye, se publican bajo la licencia MIT de *software* libre.
- La documentación del proyecto también es libre e incluye varios libros y decenas de tutoriales específicos.
- Aprender a programar con Symfony permite acceder a una gran variedad de proyectos: el *framework* Symfony2.8 para crear aplicaciones complejas, el micro *framework* Silex para sitios web sencillos y los componentes Symfony para otras aplicaciones PHP.
- Los componentes de Symfony son tan útiles y están tan probados, que proyectos tan gigantescos como Drupal 8 están contruidos con ellos.
- Para evitar el uso de contraseñas en archivos de configuración, Symfony permite establecer los parámetros de configuración de las aplicaciones a través de variables de entorno del propio servidor.
- En producción, las aplicaciones Symfony solamente necesitan permiso de escritura en dos directorios internos de la propia aplicación. Además, Symfony incluye varias herramientas gráficas y de consola para depurar fácilmente los errores que se produzcan en las aplicaciones.

Esta herramienta fue seleccionada pues es fácil de instalar y configurar en cualquier plataforma. Las aplicaciones desarrolladas con Symfony son compatibles con la mayoría de las plataformas, bibliotecas e infraestructuras que existen. Se adaptan a entornos de negocio en cambio permanente, requiriendo menos esfuerzo para su mantenimiento. También si alguna vez se encuentran dificultades, se puede contar con la colaboración de una comunidad de cientos de miles de programadores y con la seguridad de que cualquier posible defecto será corregido en versiones posteriores.

---

<sup>4</sup> Secure Digital(SD), formato de tarjeta de memoria flash.

### **1.7.7 Servidor de aplicaciones web**

**Apache 2.4:** es un servidor web multiplataforma de código abierto. Es modular (basado en módulos), donde cada módulo ofrece un grupo de funcionalidades específicas al servidor. Es uno de los servidores web más utilizado en internet, lo que facilita el acceso a la documentación. Provee un alto nivel de seguridad y eficiencia, permitiendo además el uso de una versión local, la cual hace posible que el servidor actúe como servidor y cliente al mismo tiempo, creando así la posibilidad de pre visualizar y probar el código mientras este es desarrollado (32).

Esta herramienta fue seleccionada pues es altamente configurable para el desarrollo del sistema a implementar, además de que tiene una amplia aceptación en la red. También posee código abierto y es fácil de conseguir ayuda/soporte.

## **1.8 Lenguaje de programación**

**PHP 7.1:** es un lenguaje interpretado con una sintaxis similar a la de C++ o Java. Aunque el lenguaje se puede usar para realizar cualquier tipo de programa, es en la generalización dinámica de páginas web donde ha alcanzado su máxima popularidad. En concreto, suele incluirse incrustado en páginas HTML (o XHTML), siendo el servidor web el encargado de ejecutarlo. Algunas de las características de su enorme popularidad son (33):

- Es un lenguaje libre.
- Está disponible para muchos sistemas (GNU/Linux, Windows, UNIX).
- Tiene una extensa documentación oficial en varios idiomas.
- Existen multitud de extensiones: para conectar con bases de datos, para manejo de *sockets*, para generar documentos PDF, para generar dinámicamente páginas en *flash*.
- Al ejecutarse en el servidor, los programas PHP lo pueden usar todo tipo de máquinas con todo tipo de sistemas operativos.
- En caso de que un cliente falle (por error hardware, virus) se puede seguir usando el sistema desde otro cualquiera que tenga un navegador web con conexión al servidor.

Debido a las características que presenta este lenguaje mencionadas anteriormente, será utilizado para la implementación del sistema del lado del servidor.

**C++:** es un lenguaje imperativo orientado a objetos derivado del C. En realidad, un superconjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro sigue manteniendo una considerable potencia para programación a bajo nivel, se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción. Pero a efectos prácticos,

dado el esfuerzo de compatibilidad desplegado en su diseño, puede considerarse que C++ es una extensión del C clásico (34).

Las principales ventajas de C++ son (35):

- Los programas nuevos pueden ser desarrollados en menos tiempo porque se puede rehusar el código.
- El manejo de memoria en C++ es más fácil y transparente.
- Los programas tendrán menos errores porque C++ usa una sintaxis y chequeo de tipos más estricto.
- Orientación a objetos.
- Sobrecarga de operadores y funciones.
- Rapidez.
- Genera programas compactos.
- Argumentos de funciones por default.

Este lenguaje de programación fue seleccionado pues el lenguaje de las herramientas arduino y *ethernet shield* está basado en c++, siendo así la implementación del sistema del lado del cliente.

### 1.9 Entorno Integrado de Desarrollo(IDE)

Un entorno de desarrollo integrado, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic. Algunos ejemplos de IDE son los siguientes: PhpStorm, Eclipse, Qtcreator, NetBeans, Visual C++, Arduino (36).

A continuación, se hace una breve descripción de los IDE que serán usados para el desarrollo del sistema a implementar.

**PhpStorm 8.0:** Es un IDE de programación de entornos más completos de la actualidad pues permite la gestión de proyectos fácilmente, proporciona un fácil autocompletado de código. PhpStorm ofrece editor de código inteligente con resaltado de sintaxis, la configuración de formato de código extendido, sobre la marcha de comprobación de errores, y la finalización de código inteligente (37).

**Arduino-1.6.7:** Es un entorno integrado de desarrollo para el *software* de código abierto, este se ejecuta en Windows, Mac OS X y Linux. El entorno está escrito en Java y está basado en *Processing* y otro *software* de código abierto, este se puede usar con cualquier placa arduino. Los programas de arduino están

compuestos por un solo fichero con extensión “ino”, aunque es posible organizarlo en varios ficheros. La principal característica del (IDE) y del lenguaje de programación es su sencillez y facilidad de uso (38).

## 1.10 Sistema gestor de base de datos

Un gestor de base de datos (*DataBase Management System*) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de la información del modo más eficiente posible (39).

### 1.10.1 Tipos de sistemas gestores de bases de datos

Como elemento importante de esta investigación, se realizará un estudio y caracterización de algunos sistemas gestores de bases de datos para posteriormente seleccionar el más indicado a utilizar en la propuesta de solución, dentro de los que se encuentran los siguientes (39):

**MySQL:** es un sistema de gestión de base de datos, multihilo y multiusuario seguramente el más usado en aplicaciones creadas como *software* libre. Por un lado, se ofrece bajo la GNU GPL<sup>5</sup>, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

Ventajas:

- Velocidad al realizar las operaciones.
- Bajo costo en requerimientos para la elaboración de bases de datos.
- Facilidad de configuración e instalación.

**Microsoft SQL Server:** es un sistema de gestión de bases de datos relacionales basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Es un sistema propietario de *Microsoft*. Sus principales características son:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

---

<sup>5</sup> Licencia Pública General es la licencia de derecho de autor más ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

- Además, permite administrar información de otros servidores de datos.

Su principal desventaja es el precio, aunque cuenta con una versión *EXPRESS* que permite usarlo en entornos pequeños. (Aproximadamente unos 4GB de información y varios millones de registros por tabla).

**Oracle:** es un sistema de gestión de base de datos relacional (o *RDBMS* por el acrónimo en inglés de *Relational Data Base Management System*), fabricado por *Oracle Corporation*.

Tradicionalmente Oracle ha sido por excelencia considerado siempre como el más completo y robusto, destacando por:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

También siempre ha sido considerado de los más caros, por lo que no se ha estandarizado su uso como otras aplicaciones.

Al igual que SQL Server, Oracle cuenta con una versión *EXPRESS* gratis para pequeñas instalaciones o usuarios personales.

**Microsoft Access:** es un sistema de gestión de bases de datos Relacional creado por *Microsoft (DBMS)* para uso personal de pequeñas organizaciones. Se ha ofrecido siempre como un componente de la *suite Microsoft Office*, aunque no se incluye en el paquete “básico”.

Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas.

Entre las principales funcionalidades reseñables se pueden indicar que:

- Permite crear tablas de datos indexadas.
- Modificar tablas de datos.
- Relaciones entre tablas (creación de bases de datos relacionales).
- Creación de consultas y vistas.
- Consultas referencias cruzadas.
- Consultas de acción (*INSERT, DELETE, UPDATE*).
- Formularios.

- Informes.
- Entorno de programación a través de VBA.
- Llamadas a la *API* de *Windows*.

**PostgreSQL:** es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de *PostgreSQL* no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y apoyada por organizaciones comerciales. La comunidad *PostgreSQL* se denominada el *PGDG (PostgreSQL Global Development Group)*.

Sus principales características son:

- Alta concurrencia: mediante un sistema denominado *MVCC* (Acceso concurrente multiversión, por sus siglas en inglés).
- Amplia variedad de tipos nativos: provee nativamente varios soportes.
- Ahorros considerables de costos de operación.
- Estabilidad y confiabilidad.

**DB2:** este SGBD es propiedad de *IBM*, bajo la cual se comercializa el sistema de gestión de base de datos. Utiliza XML como motor, además el modelo que utiliza es el jerárquico en lugar del modelo relacional que utilizan otros gestores de bases de datos.

Sus características más importantes son:

- Permite el manejo de objetos grandes (hasta 2 GB).
- La definición de datos y funciones por parte del usuario, el chequeo de integridad referencial.
- SQL recursivo, soporte multimedia: texto, imágenes, video, audio; *queries* paralelos, *commit* de dos fases, *backup/recuperación online* y *offline*.
- Permite agilizar el tiempo de respuestas de esta consulta.
- Recuperación utilizando accesos de sólo índices.
- Predicados correlacionados.
- Tablas de resumen.
- Tablas replicadas.
- Uniones *hash*.

Su principal desventaja es el precio, está dirigido solo a grandes empresas con necesidades de almacenamiento y procesamiento muy altas. Al igual que *SQL Server* y *Oracle* dispone de una versión *EXPRESS* gratis pero no de libre distribución.

### **1.10.2 Selección del sistema gestor de base de datos a utilizar**

Después de haber analizado las características de los diferentes sistemas gestores de bases de datos debido a que es necesario almacenar la información de todas las personas autorizadas en las diferentes áreas o locales en el sistema que se va a desarrollar, se decidió usar MySQL en su versión 5.6 pues tiene un alto rendimiento probado, fiabilidad, alta velocidad para consultas y facilidad de uso con el servidor apache a través del panel de control *xampp*. También se ha convertido en la base de datos líder elegida para las aplicaciones basadas en web.

### **1.10.3 PhpMyAdmin 2.2.6 para el manejo de MySQL**

PhpMyAdmin es una herramienta de *software* libre escrita en PHP, destinada a manejar la administración de MySQL a través de la web. Es compatible con una amplia gama de operaciones en MySQL y MariaDB. Las operaciones de uso frecuente (administración de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos) se pueden realizar a través de la interfaz de usuario, mientras que todavía se tiene la capacidad de ejecutar directamente cualquier declaración de SQL (40).

Características de PhpMyAdmin (40):

- Interfaz web intuitiva.
- Soporta la mayoría de las características de MySQL:
- Examinar y eliminar bases de datos, tablas, vistas, campos e índices.
- Crear, copiar, eliminar, cambiar el nombre y alterar bases de datos, tablas, campos e índices.
- Mantenimiento del servidor, bases de datos y tablas, con propuestas sobre la configuración del servidor.
- Ejecutar, editar y marcar cualquier sentencia SQL.
- Administrar cuentas de usuario de MySQL y privilegios.
- Gestionar procedimientos almacenados y disparadores.

## **1.11 Relación de costo**

A lo largo de todos los años en que arduino existe, desde 2006, han sido muy numerosas los diferentes tipos de placa que la plataforma ha sacado al mercado de forma directa. Actualmente la oferta de placas arduino oficiales es de 12 modelos y hay otras tantas ya descatalogadas. A ellas hay que sumar diferentes añadidos, ampliaciones y también kits que vende arduino a través de su web o distribuidores oficiales (49).

En la tabla 2 se pueden apreciar varios modelos, la cantidad de entradas y salidas digitales, así como las entradas analógicas y la memoria de almacenamiento interno de estos dispositivos, también el precio en euros (€) en el mercado internacional.

Modelo	Entradas/salidas digitales	Entradas analógicas	Memoria	Precio
Uno r3	16	6	32kb	7,5€
Leonardo	20	12	32kb	10€
Mega r3	54	16	256kb	12,5€
Mini 05	14	6	32kb	2,5€

Tabla 2: Modelos de arduino

Fuente:(49)

Arduino uno	<i>Ethernet shield</i>	Módulo Rc522 <i>RFID</i> con 2 tarjetas <i>RFID</i>	Total
7,5€	4,61€	6€	18,1€

Tabla 3: Costo del prototipo a utilizar

Fuente:(50)

## Conclusiones del capítulo

La evidencia que se expuso anteriormente demuestra que en el presente capítulo se realizó un estudio de los conceptos asociados con el control, control de acceso y sistemas de control de acceso, lo que ayudó a obtener un mejor entendimiento de los mismos dentro del marco teórico de la investigación. Para la realización de la solución se seleccionó la metodología de desarrollo de *software* AUP-UCI, pues es recomendada por la Universidad de Ciencias Informáticas para el desarrollo de proyectos productivos. El análisis de los tipos de tecnologías de control de acceso contribuyó a definir a la *RFID* como la más apropiada a utilizar por tener ventajas sobre los códigos de barras, tarjetas magnéticas y sistemas biométricos. Debido a que se necesita el envío de datos hacia un servidor web, se selecciona el *ethernet shield* como herramienta para conectar un arduino a una red cableada, siendo estas mismas placas de hardware libre y de uso fácil.

## Capítulo 2: Análisis y diseño de la solución propuesta.

### Introducción

En este capítulo se describirá la propuesta de solución mediante el análisis y diseño del sistema, tomando como referencia los conceptos y términos definidos en el primer capítulo, y regido por la metodología de desarrollo AUP – UCI. Se enunciarán los requisitos funcionales y no funcionales y la arquitectura que será seleccionada, así como los artefactos generados en la etapa correspondiente al modelado del sistema.

### 2.1 Propuesta de solución

Para dar solución a la situación problemática, se propone implementar un nuevo sistema de control de acceso, utilizándose la tecnología *RFID*. Este sistema tiene como principal objetivo establecer un control del personal que accede a un determinado local o área restringida dentro de la universidad, así como monitorear la entrada y salida a los mismos. Existen dos vías para poner en práctica dicha solución:

1. La utilización de cerraduras *RFID* con arduino. (ver figura 1)
2. La utilización de personas para controlar el acceso físicamente utilizando también arduino y *RFID*.

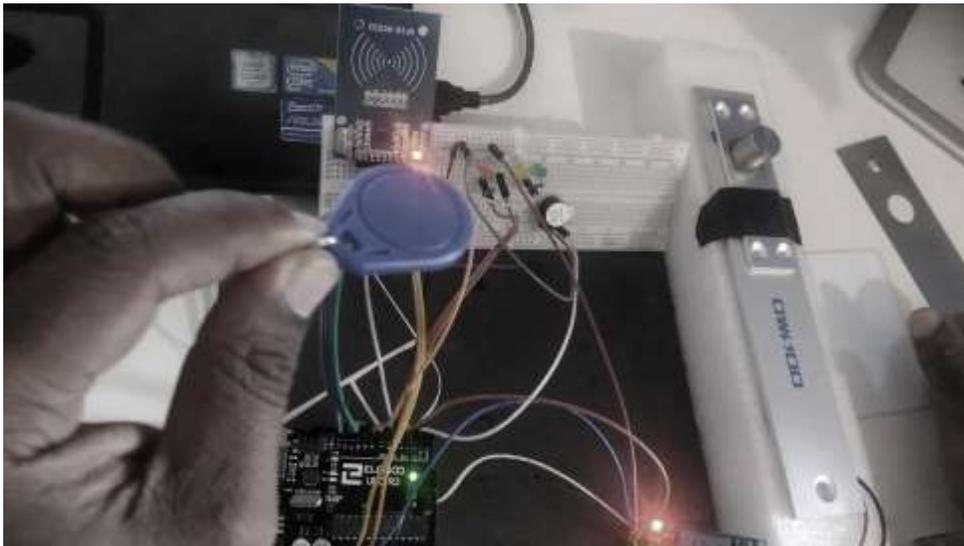


Figura 1: Cerraduras *RFID*

Fuente: (41)

Debido a que en la UCI no se cuenta con la tecnología de las cerraduras *RFID*, se propone utilizar la segunda vía mencionada anteriormente.

Para ello se propone realizar una aplicación web funcionando como servidor en una PC que estará en el local del administrador del sistema; a esta PC estarán conectados mediante la red cableada todos los arduinos con su *ethernet shield* pertenecientes a cada local que se pretende controlar el acceso del personal

hacia ellos. Los arduinos estarán funcionando como clientes y se propone que tengan baterías conectadas, al igual que la PC servidor tenga un *Backus* funcionando las 24 horas en caso de que falle la corriente eléctrica para que no se afecte el funcionamiento del sistema.

El administrador del sistema tendrá los privilegios de gestionar las personas y locales en el sistema, así como exportar reportes en formato PDF sobre los accesos realizados diariamente. Los directivos de la universidad y el administrador serán los únicos que tendrán acceso a todos los locales existentes en el sistema independientemente según el área a la que pertenecen. Se podrá registrar tarjetas, o sea, almacenarlas en la base de datos y asignársela a personas, uno o varios locales y asignarles permisos a diferentes tarjetas para controlar el acceso a áreas restringidas y la entrada y salida a los locales en la UCI. Esto será posible utilizando una placa de *hardware* libre, en este caso, un arduino que mediante la interfaz de comunicación SPI se puedan enviar datos desde el módulo lector *RFID* hacia el arduino y este mediante el *ethernet shield* al sistema y posteriormente poder definir si una persona es autorizada o no. También se establecerá una conexión entre el servidor web y la base de datos mediante el protocolo TCP para obtener toda la información referente a los sujetos que accedieron a los distintos locales de la universidad.

Los técnicos para poder monitorear la entrada y salida a los locales en la UCI tendrán una PC, mediante la cual podrán también comprobar que las tarjetas pertenecen a los individuos que intentan acceder a un determinado local y para tener una representación más fácil de saber si una persona es autorizada o no, estos se podrán guiar por los bombillos led que están conectados al arduino al encender después que se reciba la respuesta del servidor al pasar las identificaciones por el módulo lector *RFID*, amarillo autorizado y rojo no autorizado. Así como también podrán visualizar el reporte de acceso de las personas para tener un control detallado según la fecha y hora de entrada y salida a los locales. Los mismos solo podrán acceder a la página de los reportes de accesos. A continuación, para un mejor entendimiento, se mencionan los roles que van a interactuar con el sistema a desarrollar y su descripción (ver Tabla 2)

Tabla 4: Descripción de los roles del sistema

Rol	Descripción
Administrador del sistema	Representa a la persona que se va a encargar de administrar el sistema.
Técnico	Representa a la persona que se encargará de monitorear la entrada y salida a los locales y de controlar el acceso a los mismos.

Fuente: elaboración propia

### **2.1.1 Diagrama de flujo del sistema**

A partir de la propuesta de solución planteada anteriormente se decide realizar un diagrama de flujo para mostrar el flujo de control general sobre los aspectos operacionales de las actividades de trabajo en el sistema a desarrollar.

Es un diagrama que representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Sirve para dos propósitos (42):

- Indica como se transforman los datos conforme avanza el sistema.
- Representa las funciones (y subfunciones) que transforman el flujo de datos.

A continuación, se realiza un diagrama de flujo para dar solución al problema de la investigación con una breve descripción.

Al intentar acceder una persona a un determinado local, si el sistema lee la etiqueta, se verifica si está autorizada o no, en caso de que no se lea el administrador se encarga de verificar el problema existente. Si esta no es autorizada, se muestra el id de dicha etiqueta y el acceso sería denegado. Si la misma es autorizada, se registra el acceso y por consiguiente la persona puede acceder al local.

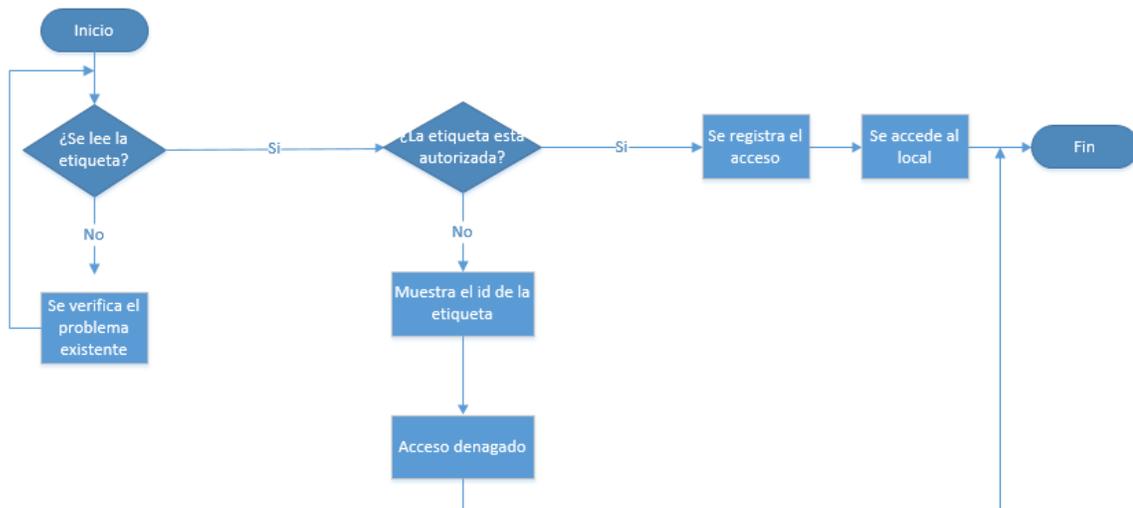


Figura 2: Diagrama de flujo de la propuesta de solución  
Fuente: elaboración propia

## 2.2 Requisitos de Software

Los requisitos de *software* constituyen las necesidades de los clientes, las funcionalidades y las restricciones que debe cumplir el sistema. Los mismos se clasifican en funcionales y no funcionales (42).

### 2.2.1 Requisitos Funcionales

Los Requisitos Funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de como se debe comportar en situaciones particulares. En algunos casos, los RF de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (43). A continuación, se identifican los RF del sistema a desarrollar.

RF 1. Insertar datos de persona

RF 2. Modificar datos de persona

RF 3. Eliminar persona

RF 4. Eliminar múltiples personas

RF 5. Listar persona

RF 6. Registrar tarjeta

RF 7. Modificar estado de tarjeta

RF 8. Eliminar tarjeta

RF 9. Eliminar múltiples tarjetas

RF 10. Listar tarjeta

RF 11. Reportar accesos

RF 12. Obtener datos de persona dado usuario

RF 13. Asignar grupo de acceso a persona

RF 14. Acceder a local

RF 15. Comprobar acceso a local

RF 16. Notificar.

RF 17. Insertar datos de local

RF 18. Modificar datos de local

RF 19. Listar local

RF 20. Eliminar local

RF 21. Eliminar múltiples locales

### **2.2.2 Requisitos No Funcionales**

Los Requisitos No Funcionales (RNF), como su nombre sugiere, son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (43). De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema. Los RNF rara vez se asocian con características particulares del sistema; más bien, estos requisitos especifican o restringen las propiedades emergentes del sistema.

El sistema debe cumplir con los siguientes RNF agrupados según su clasificación:

#### **RNF 1. Interfaz de usuario:**

El sistema debe tener indicadores que permitan conocer al usuario las acciones que debe realizar, por ejemplo, botones con íconos sugerentes y alternativa textual.

### **RNF 2. Usabilidad:**

Los grupos de botones y vínculos deben estar organizados por funcionalidad, con el objetivo de facilitar al usuario la interacción con el sistema.

Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos, en idioma español.

### **RNF 3. Software:**

Características del *software* en el servidor con las siguientes condiciones mínimas:

- Sistema operativo: Basados en GNU/Linux, Windows XP/7/8/10.
- Servidor web: Apache.
- PHP 7.2
- MySQL 5.6 como Sistema Gestor de Base de Datos.
- Un navegador como Mozilla Firefox v48.0, Chrome v42.0.

### **RNF 4. Hardware:**

Características del *hardware* en el servidor con las siguientes condiciones mínimas:

- Procesador: 1.8Ghz.
- Memoria RAM: 2GB.
- Disco Duro: 1GB

## **2.3 Descripción de Historias de Usuario (HU)**

Entre los artefactos que define la metodología seleccionada se encuentran las HU que son utilizadas para especificar las funcionalidades que brindará el sistema. Cada HU es una representación de un requerimiento de *software* escrito en una o dos frases utilizando el lenguaje común del usuario. Representan una forma rápida de administrar los requerimientos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos (3).

A continuación, se muestra la descripción de algunas HU, para ver las restantes ver Anexos (1, 2..., 18):

Tabla 5: HU del RF1

Número: HU 1	
Número: HU 1	Nombre del requisito: Insertar datos de persona

<b>Programador:</b> Yasiel Silva Gómez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>	<b>Tiempo Real:</b> 2 días
<b>Descripción:</b> El sistema deberá permitir insertar el área, la cual será el centro al que pertenece la persona, el nombre de la persona, los apellidos, el solapín, también la foto y el usuario los cuales serán tomados del directorio UCI, el cargo será estudiante, trabajador y visitante y el sexo de una persona.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>• Si la persona es un visitante, no será necesario insertar el área, la foto, el solapín, ni el usuario, al no ser que pertenezca a algún centro de trabajo en la UCI.</li> </ul>	
<b>Prototipo elemental de interfaz gráfica de usuario:</b>	
	

Fuente: elaboración propia

Tabla 6: HU del RF2

<b>Número:</b> HU 2	<b>Nombre del requisito:</b> Modificar datos de persona
<b>Programador:</b> Yasiel Silva Gómez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día

**Riesgo en Desarrollo:**

- Problemas de fallos de conexión.
  - Planificación incorrecta.

**Tiempo Real:** 2 días**Descripción:**

El sistema deberá permitir actualizar los datos de cada persona tales como el área, el nombre, los apellidos, el solapín, la foto y el usuario, el cargo y el sexo.

**Observaciones:**

- Si la persona es un visitante, no será necesario insertar el área, la foto, el solapín, ni el usuario.

**Prototipo elemental de interfaz gráfica de usuario:**

El prototipo muestra una interfaz con el título "Modificar persona". Incluye los siguientes campos de entrada:

- Usuario (campo de texto)
- Solapín/CI (campo de texto)
- Nombre (campo de texto)
- Apellidos (campo de texto)
- Foto (campo de texto)
- Sexo (menú desplegable)
- Cargo (menú desplegable)
- Área (campo de texto)

En la parte inferior hay tres botones: "Editar", "Cancelar" y "Eliminar".

Fuente: elaboración propia

Tabla 7: HU del RF6

<b>Número:</b> HU 6	<b>Nombre del requisito:</b> Registrar tarjeta
<b>Programador:</b> Yasiel Silva Gómez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>	<b>Tiempo Real:</b> 2 días
<b>Descripción:</b> El sistema deberá permitir agregar una nueva tarjeta al sistema insertando el número de la misma de forma manual o automática, así como la persona a la que estará asignada y el local al cual va a tener acceso y si estará habilitada o no y almacenarla en la base de datos del sistema.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>• Las tarjetas solo podrán ser asignadas a una sola persona y tendrán acceso a uno o varios locales.</li> <li>• El número de las tarjetas es en hexadecimal de 10 dígitos y no podrán existir dos tarjetas con el mismo número.</li> </ul>	

### Prototipo elemental de interfaz gráfica de usuario:

El prototipo muestra una interfaz con el título "Registrar tarjeta". Incluye un campo de texto para "Número", un checkbox para "Habilitada", un menú desplegable para "Persona", y un selector de "Locales (No permitidos <-> Permitidos)" con dos botones de selección. En la parte inferior hay dos botones: "Crear" y "Regresar".

Fuente: elaboración propia

## 2.4 Arquitectura del software

La arquitectura del *software* de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del *software*, sus propiedades externas visibles y las relaciones entre ellos. Es una representación que permite analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados con la construcción del *software* (42).

La arquitectura del *software* es importante por las siguientes razones (42):

- Las representaciones de la arquitectura del *software* permiten la comunicación entre todas las partes (participantes) interesadas en el desarrollo de un sistema basado en computadora.
- La arquitectura resalta las primeras decisiones que tendrán un efecto profundo en todo el trabajo de ingeniería de *software* siguiente y, también importante, en el éxito último del sistema como entidad operacional.
- La arquitectura constituye un modelo relativamente pequeño y asequible por la vía intelectual sobre como está estructurado el sistema y la forma en la que sus componentes trabajan juntos.

### 2.4.1 Descripción de la arquitectura del sistema

En el sistema a desarrollar se utilizará el MVC o Modelo-Vista-Controlador, este es un patrón de arquitectura de *software* que, utilizando 3 componentes (Vistas, Modelos y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Es una arquitectura importante puesto que se utiliza

tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los *frameworks* modernos utilizan MVC (o alguna adaptación del mismo) para la arquitectura, entre ellos se pueden mencionar a *Ruby*, *Django*, *Symfony*, *AngularJS* (44). *Symfony* está basado en dicho patrón arquitectónico, el cual permite que los cambios en la vista no afecten el resto de la aplicación.

El Modelo-Vista-Controlador permite dividir la lógica de negocio del diseño, haciendo el proyecto más escalable. Admite utilizar abstracción de datos, facilitando la realización de consultas a la base de datos. En el caso de proyectos donde hay varios desarrolladores, el seguir métodos comunes de programación, hace que el código sea más entendible entre estos, pudiendo uno continuar el trabajo de otro. En estos casos es conveniente utilizar herramientas de control de versiones como Subversión. Un *framework* MVC ayuda a controlar los recursos del servidor, evitando Bugs <sup>6</sup>que puedan repercutir en el rendimiento, por ejemplo, muchas veces se olvida cerrar conexiones a la base de datos, sobrecargando el servidor.

A continuación, para un mejor entendimiento, se describen los componentes de la arquitectura seleccionada.

**Modelo:** se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, todo eso va aquí, en el modelo.

**Controlador:** se encarga de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.

**Vistas:** son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se encargan de como se verán los datos, esa responsabilidad es únicamente de la vista.

---

<sup>6</sup> Para nombrar a los errores que se producen en un programa informático.

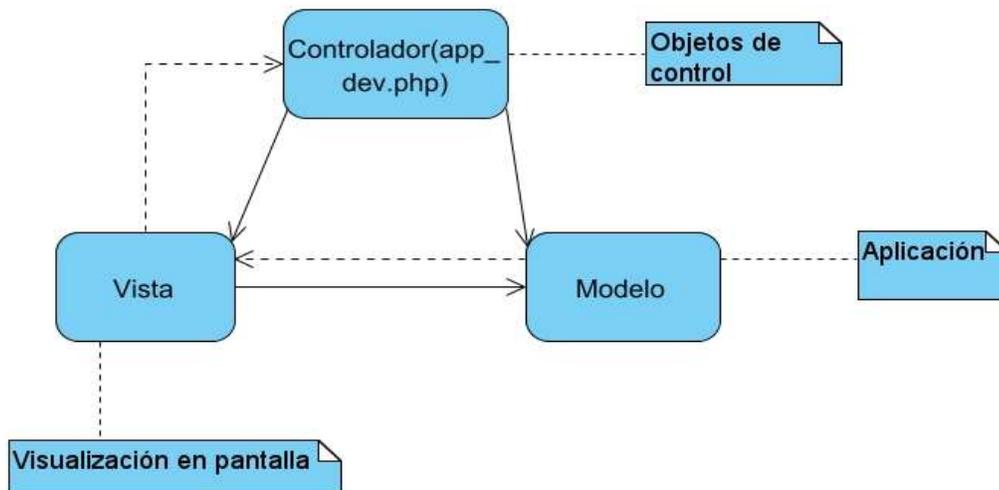


Figura 3: Patrón de diseño MVC

Fuente: elaboración propia

En la solución propuesta, la vista es la página HTML que es mostrada al usuario (index.html. twig) y el controlador es donde están implementadas las funcionalidades de la aplicación (app\_dev.php).

## 2.5 Patrones de diseño

A partir de la arquitectura de *software* seleccionada debido a que el *framework* *Symfony* la utiliza, se hace necesario utilizar patrones de diseño para la búsqueda de soluciones a problemas comunes en el desarrollo del sistema a desarrollar.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular, una solución estándar para un problema común de programación, una manera más práctica de describir ciertos aspectos de la organización de un programa. También es un modelo que permite guiar la realización de una actividad determinada, ahorran tiempo en la construcción del *software* y forman un vocabulario común entre los diseñadores. Los mismos están orientados al cambio, soportan la reutilización de arquitecturas *software*, capturan la experiencia y la hacen accesible a los no expertos (42).

Para la realización del sistema de control de acceso basado en *RFID*, se utilizarán los Patrones de Asignación de Responsabilidades (*GRASP*) y Patrones GOF (Gang of Four).

### 2.5.1 Patrones GRASP

Los patrones *GRASP* describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. *GRASP* es un acrónimo que significa (Patrones Generales de

*Software* para Asignar Responsabilidades). El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el *software* orientado a objetos (45).

**Controlador:** sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal manera que es el que recibe todas las peticiones y datos del usuario y ejecuta en las diferentes clases el método solicitado. Este patrón se aplica a la clase *app\_dev.php* que es el controlador frontal, pues se encarga de obtener la información de una petición HTTP(Request) y de generar y devolver la respuesta HTTP(Response).

**Alta Cohesión:** la cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento de un diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Este patrón se evidencia en el sistema de manera tal que las funcionalidades contienen la información necesaria para trabajar sin involucrar a otras funcionalidades.

**Bajo Acoplamiento:** tiene como objetivo que las clases del sistema tengan la menor dependencia entre ellas. De tal manera que, en caso de producirse alguna modificación, el impacto sea el mínimo posible en el resto de clases, potenciando la reutilización. Este patrón se evidencia en las clases *VisitanteController.php* y *PersonaController.php* pues los cambios que se realicen en una clase no afecta a la otra, o sea, son fáciles de entender por separados y factibles de reutilizar.

**Experto:** es el principio básico de asignación de responsabilidades, define que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Además, es la clase que contiene la información necesaria para cumplir la responsabilidad. Tal es el caso de las clases de acceso a datos que son las encargadas de toda la lógica del acceso a los datos. *Symfony* utiliza la librería Doctrine para todo lo que tiene que ver con base de datos, cada clase creada por Doctrine a partir de una entidad es experta en manejar su información.

**Creador:** este patrón es el encargado de guiar la asignación de responsabilidades relacionada con la creación de los objetos y tiene como propósito fundamental encontrar un creador que se conecte con el objeto producido en cualquier evento, es decir asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Este se evidencia en las clases *PersonaController.php*, *TarjetaController.php*, *LocalController.php* y *AccesoController.php*. En estas clases las acciones se encargan de crear los objetos

de las clases que representan las entidades, evidenciando de este modo que cada clase controladora es el “creador” de las entidades.

### **2.5.2 Patrones GOF**

Los patrones GOF describen soluciones simples y elegantes a problemas específicos en el diseño de *software* orientado a objetos, además tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad (42).

**Decorador:** este patrón se evidencia en la clase abstracta Views, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo index.html.twig, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas del sistema para no tener que repetirlo en cada página.

**Registro:** este patrón es muy útil para los desarrolladores en la programación orientada a objetos. Este es un medio sencillo y eficiente de compartir datos y objetos en la aplicación sin la necesidad de preocuparse por conservar numerosos parámetros o hacer uso de variables globales. El mismo se aplica en la clase Config, que es la encargada de acumular todas las variables de uso global en el sistema.

### **2.6 Modelo de datos**

Un modelo de datos se puede definir como un conjunto de conceptos, reglas y convenciones bien definidos que permiten aplicar una serie de abstracciones a fin de describir y manipular los datos de un cierto mundo real que se desea almacenar en la base de datos. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos (43).

Por ello, se le considera fundamental para el desarrollo de cualquier aplicación que necesite almacenar datos. En la Figura 4 se presenta el diagrama correspondiente al modelo entidad-relación de base de datos utilizada para el manejo de información del sistema propuesto.

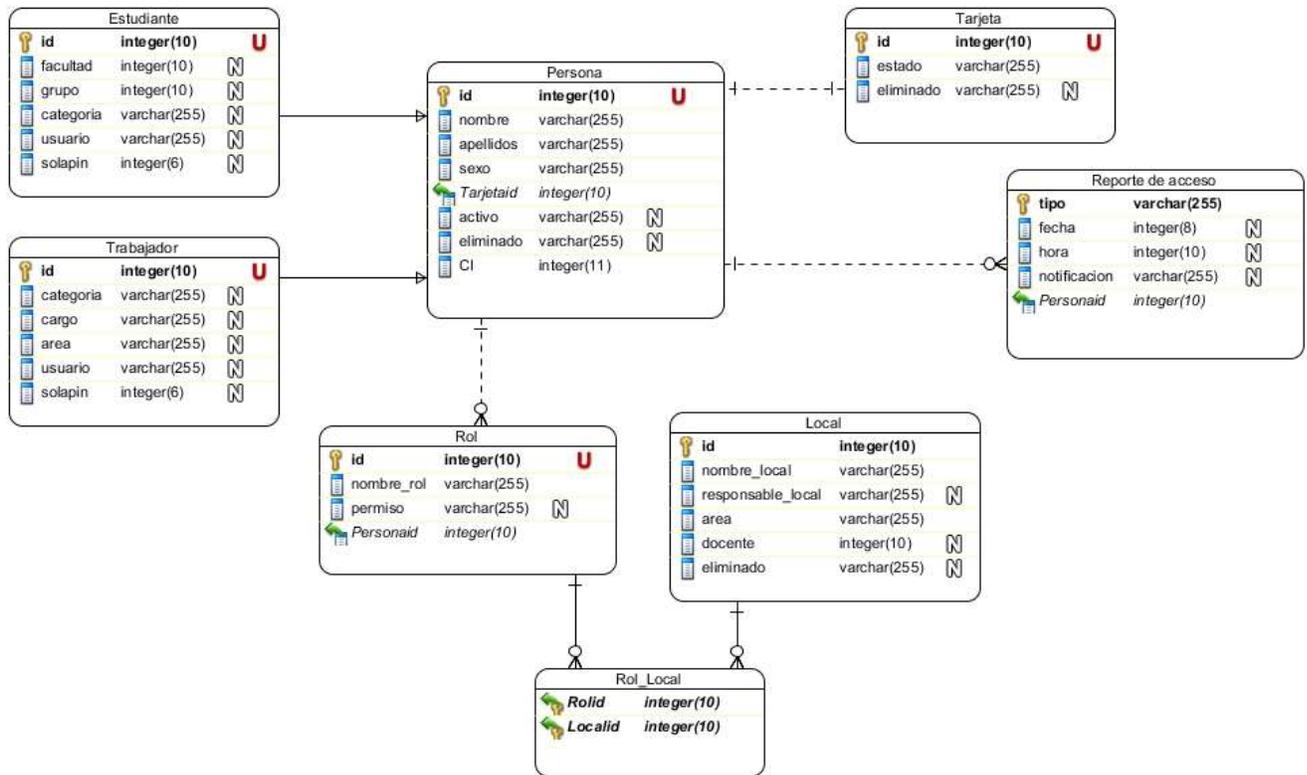


Figura 4: Modelo Entidad-Relación

Fuente: elaboración propia

## Conclusiones del capítulo

En el presente capítulo quedó expresada la etapa de análisis y diseño de la solución propuesta, arrojando como resultado un grupo de artefactos para organizar y entender los principales requerimientos del sistema, descritos en sus respectivas historias de usuarios. Se obtuvo como resultado el diagrama entidad relación que define el diseño de la base de datos. Se estableció como patrón arquitectónico el MVC para facilitar la realización de consultas a la base de datos. El empleo adecuado de los patrones arquitectónicos Experto, Controlador, Alta Cohesión, Bajo Acoplamiento, Creador, Decorador y Registro posibilitó realizar una correcta implementación del sistema propuesto.

## Capítulo 3: Implementación y pruebas

### Introducción

En el presente capítulo se mostrarán los estándares de codificación empleados en el desarrollo del sistema, se expresarán los artefactos ingenieriles relacionados con la implementación y validación del sistema a desarrollar. Entre los principales elementos se encuentran el modelo de implementación y el de despliegue, con los respectivos artefactos generados a partir de ellos. Además, se especificarán las pruebas realizadas al sistema donde se evidenciará la evaluación de la calidad del producto realizado.

### 3.1 Estándares de código

Las técnicas de codificación incorporan muchos aspectos del desarrollo del software. Aunque generalmente no afectan a la funcionalidad de la aplicación, sí contribuyen a una mejor comprensión del código fuente. En esta fase se tienen en cuenta todos los tipos de código fuente, incluidos los lenguajes de programación, de secuencias de comandos, de marcado o de consulta (46). Para la realización del sistema se definieron las siguientes pautas de codificación:

✓ Nombres:

1. Las clases, interfaces y similares comenzarán con letra inicial mayúscula. En el caso de las clases controladoras seguido del nombre de la clase se utiliza la palabra Controller pues el marco de trabajo Symfony propone este estándar de codificación para las clases de este tipo.

Ejemplo: `class PersonaController extends Controller`

2. Los nombres de los métodos deben comenzar con letra inicial minúscula y a continuación la palabra Action, en caso de ser compuestos la próxima palabra debe iniciar con mayúscula.

Ejemplo: `public function indexAction()`

3. Para los identificadores de variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto la próxima palabra debe iniciar con mayúscula.

Ejemplo: `$tarjetas = $em->getRepository('TarjetaBundle:Tarjeta')->findAll()`

### 3.2 Modelo de implementación

El modelo de implementación permite representar como se implementan en términos de componentes, los elementos del modelo de diseño. Describe también como dependen los componentes unos de otros, y como se organizan de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizado (47). Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

### 3.2.1 Diagrama de componentes

El diagrama de componentes muestra la vista física de la aplicación a través de componentes y sus relaciones; representa cómo un sistema de *software* es dividido en elementos y muestra las dependencias entre estos elementos. A continuación, se presenta el diagrama de componentes correspondiente al caso de uso gestionar persona (Ver Figura 5). Para ver los restantes consultar Anexos (19 al 22).

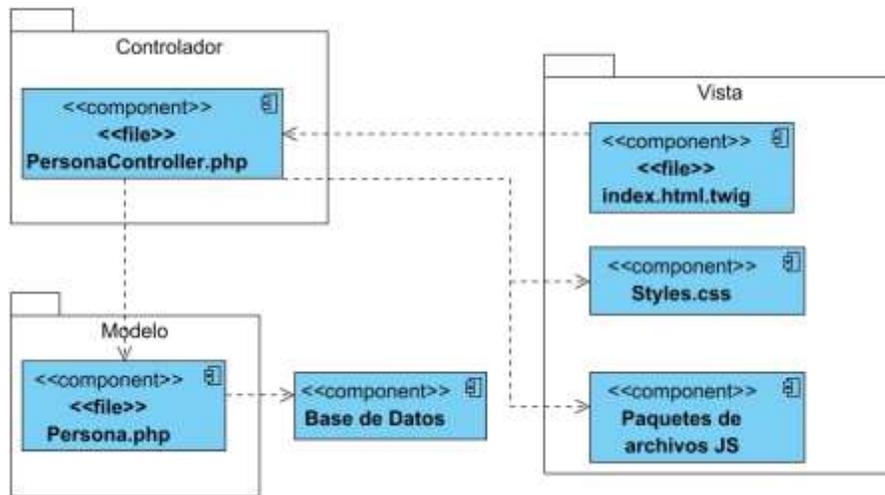


Figura 5: Diagrama de componentes gestionar persona

Fuente: elaboración propia

### 3.3 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo (47). En general dicho modelo representa la estructura del sistema en ejecución.

Con el objetivo de representar los recursos de cómputo que necesita el sistema para su correcto funcionamiento se muestra a continuación el diagrama de despliegue (Ver Figura 6) correspondiente a la solución propuesta.

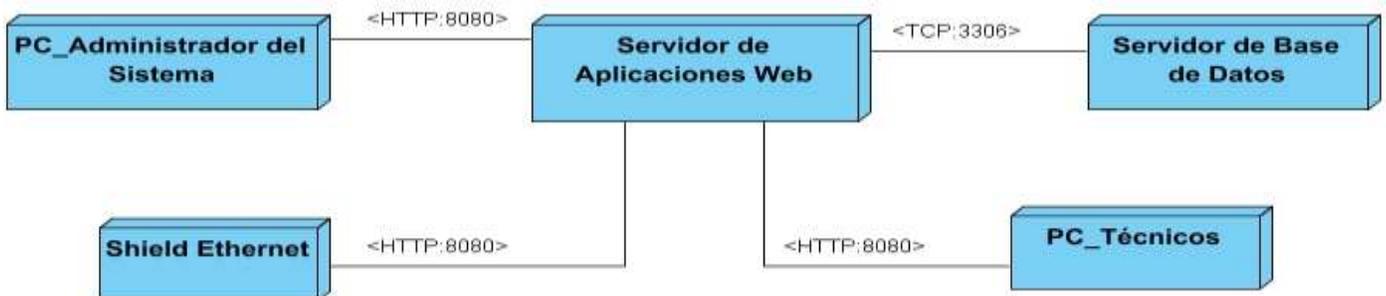


Figura 6: Diagrama de despliegue

Fuente: elaboración propia

### **3.3.1 Descripción del diagrama de despliegue**

El sistema se desplegará en la UCI, donde se debe contar al menos con una PC para el administrador del sistema y un *ethernet shield* unido a un arduino para los técnicos o personas encargadas en cada local de mantener un control de acceso eficiente, además los técnicos que estén en las puertas principales deben contar con una PC para monitorear la entrada y salida a los locales. Dichas PC y *ethernet shield* se deben conectar al servidor de aplicaciones web usando el protocolo HTTP<sup>7</sup> por el puerto 8080. Este a su vez se comunica con el servidor de base de datos utilizando el protocolo TCP<sup>8</sup> por el puerto 3306.

### **3.4 Pruebas de software**

Una vez finalizada la implementación del sistema, se hace necesario comprobar que su funcionamiento sea correcto, verificando que las funcionalidades se ajusten a las especificaciones planteadas. Para ello se realizan las pruebas de *software*, con el fin de detectar defectos y asegurar que estos sean corregidos antes de la entrega del producto al cliente.

Las pruebas son actividades en las cuales un sistema o componente es ejecutable bajo condiciones o requerimientos específicos permitiendo que los resultados sean observados y registrados, estas se realizan con el objetivo de encontrar deficiencias existentes en el *software* (42).

Durante el flujo de trabajo de pruebas se verifica el resultado final de la implementación, probando la estructura; tanto en la construcción interna como intermedia, así como las versiones finales.

#### **3.4.1 Pruebas de Aceptación**

Las pruebas de aceptación del usuario es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el *software* está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el *software* fue construido (42).

#### **Pruebas de caja negra**

Estas pruebas se llevan a cabo sobre la interfaz de la aplicación, concentrándose completamente en los requisitos funcionales del sistema e ignorando la estructura interna del software. Estos casos de pruebas pretenden demostrar que las funciones de la aplicación son operativas y que la entrada y salida se realicen de forma correcta. Las pruebas de caja negra tratan de encontrar errores en las siguientes categorías (42):

- Funciones incorrectas o faltantes.

---

<sup>7</sup> Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto)

<sup>8</sup> Transmission Control Protocol o TCP (en español protocolo de control de transmisión)

- Errores de interfaz.
- Errores de estructura de datos o en acceso a base de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización y término.

Para seleccionar el conjunto de entradas con el que trabajar, hay que tener en cuenta que existen algunos que causan comportamientos erróneos en el sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos, teniendo como objetivo final encontrar una colección de datos cuya probabilidad de pertenecer a dicho grupo sea la más alta posible (48).

Para confeccionar los casos de prueba existen distintos criterios:

- Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*.
- Técnica del Análisis de Valores Límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Para realizar las pruebas de caja negra se eligió técnica de la partición de equivalencia, la cual permite realizar un conjunto de pruebas y obtener el máximo de los errores presentes en el proceso.

Tabla 8: Caso de prueba aceptación 1

Caso de Prueba Aceptación	
<b>Código:</b> 1	<b>Historia de usuario:</b> 1
<b>Nombre:</b> Insertar datos de persona.	
<b>Descripción:</b> Prueba para la funcionalidad de insertar datos de persona.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Al seleccionar la opción "Crear una nueva persona" el sistema muestra un panel que brinda la posibilidad de insertar los datos de una persona o de un local.</li> <li>• Al seleccionar el botón "Crear" el sistema almacena los datos de una nueva persona en la base de datos.</li> </ul>	

<b>Resultado Esperado:</b> Se ha añadido la nueva persona en el sistema.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

Fuente: elaboración propia

Tabla 9: Caso de prueba aceptación 2

Caso de Prueba Aceptación	
<b>Código:</b> 2	<b>Historia de usuario:</b> 12
<b>Nombre:</b> Obtener datos de persona dado usuario.	
<b>Descripción:</b> Prueba para la funcionalidad de obtener datos de persona dado usuario.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> <li>• La PC servidor debe estar conectada a la red UCI.</li> <li>• La persona a crear debe ser estudiante o trabajador.</li> <li>• El usuario debe ser real, o sea, que esté en la base de datos de la UCI.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Al insertar el usuario de una persona y presionar la tecla "Tab" el sistema debe llenar los restantes campos vacíos correspondientes a los datos de la persona.</li> </ul>	
<b>Resultado Esperado:</b> Se han obtenido los datos de la persona dado su usuario.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 10: Caso de prueba aceptación 3

Caso de Prueba Aceptación	
<b>Código:</b> 3	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Modificar datos de persona.	
<b>Descripción:</b> Prueba para la funcionalidad de modificar datos de persona.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Al seleccionar la opción "editar" el sistema muestra un panel que brinda la posibilidad de editar los datos de una persona.</li> </ul>	

<b>Resultado Esperado:</b> Se han editado los datos de la persona en el sistema.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

Fuente: elaboración propia

Tabla 11: Caso de prueba aceptación 4

Caso de Prueba Aceptación	
<b>Código:</b> 4	<b>Historia de usuario:</b> 3
<b>Nombre:</b> Eliminar persona.	
<b>Descripción:</b> Prueba para la funcionalidad de eliminar persona.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después de marcar la casilla, al seleccionar la opción "Eliminar" el sistema muestra un panel que brinda la posibilidad de eliminar a una persona.</li> </ul>	
<b>Resultado Esperado:</b> Se ha eliminado a la persona en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 12: Caso de prueba aceptación 5

Caso de Prueba Aceptación	
<b>Código:</b> 5	<b>Historia de usuario:</b> 4
<b>Nombre:</b> Eliminar múltiples personas.	
<b>Descripción:</b> Prueba para la funcionalidad de eliminar múltiples personas.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después de marcar las casillas, al seleccionar la opción "Eliminar" el sistema muestra un panel que brinda la posibilidad de eliminar a varias personas.</li> </ul>	
<b>Resultado Esperado:</b> Se han eliminado a las personas en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 13: Caso de prueba aceptación 6

Caso de Prueba Aceptación	
<b>Código:</b> 6	<b>Historia de usuario:</b> 5
<b>Nombre:</b> Listar persona.	
<b>Descripción:</b> Prueba para la funcionalidad de listar persona.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Al seleccionar el elemento “Personas”, el sistema muestra un panel con un listado de los datos de cada persona.</li> </ul>	
<b>Resultado Esperado:</b> Se han mostrado los datos de las personas en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 14: Caso de prueba aceptación 7

Caso de Prueba Aceptación	
<b>Código:</b> 7	<b>Historia de usuario:</b> 6
<b>Nombre:</b> Registrar tarjeta.	
<b>Descripción:</b> Prueba para la funcionalidad de registrar tarjeta.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Al seleccionar la opción “Registrar tarjeta”, el sistema muestra un panel que brinda la posibilidad de crear una tarjeta de forma manual, asignarle a una determinada persona y a uno o varios locales a los cuales dicha tarjeta va a tener acceso.</li> </ul>	
<b>Resultado Esperado:</b> Se ha registrado la tarjeta en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 15: Caso de prueba aceptación 8

Caso de Prueba Aceptación
---------------------------

<b>Código:</b> 8	<b>Historia de usuario:</b> 7
<b>Nombre:</b> Modificar estado de tarjeta.	
<b>Descripción:</b> Prueba para la funcionalidad de modificar datos de tarjeta.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Al seleccionar la opción "editar" el sistema muestra un panel que brinda la posibilidad de editar los datos de una tarjeta.</li> </ul>	
<b>Resultado Esperado:</b> Se han editado los datos de la persona en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 16: Caso de prueba aceptación 9

Caso de Prueba Aceptación	
<b>Código:</b> 9	<b>Historia de usuario:</b> 8
<b>Nombre:</b> Eliminar tarjeta.	
<b>Descripción:</b> Prueba para la funcionalidad de eliminar tarjeta.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después de marcar la casilla, al seleccionar la opción "Eliminar" el sistema muestra un panel que brinda la posibilidad de eliminar a una tarjeta.</li> </ul>	
<b>Resultado Esperado:</b> Se ha eliminado a la tarjeta en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 17: Caso de prueba aceptación 10

Caso de Prueba Aceptación	
<b>Código:</b> 10	<b>Historia de usuario:</b> 9
<b>Nombre:</b> Eliminar múltiples tarjetas.	
<b>Descripción:</b> Prueba para la funcionalidad de eliminar múltiples tarjetas.	

<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>
<p><b>Respuesta del Sistema/Flujo Central:</b></p> <ul style="list-style-type: none"> <li>• Después de marcar las casillas, al seleccionar la opción "Eliminar" el sistema muestra un panel que brinda la posibilidad de eliminar a varias tarjetas.</li> </ul>
<p><b>Resultado Esperado:</b> Se han eliminado a las tarjetas en el sistema.</p>
<p><b>Evaluación de la Prueba:</b> Prueba satisfactoria.</p>

Fuente: elaboración propia

Tabla 18: Caso de prueba aceptación 11

Caso de Prueba Aceptación	
<b>Código:</b> 11	<b>Historia de usuario:</b> 10
<b>Nombre:</b> Listar tarjetas.	
<b>Descripción:</b> Prueba para la funcionalidad de listar tarjetas.	
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<p><b>Respuesta del Sistema/Flujo Central:</b></p> <ul style="list-style-type: none"> <li>• Al seleccionar el elemento "Tarjetas", el sistema muestra un panel con un listado de los datos de cada persona.</li> </ul>	
<b>Resultado Esperado:</b> Se han mostrado los datos de las tarjetas en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 19: Caso de prueba aceptación 12

Caso de Prueba Aceptación	
<b>Código:</b> 12	<b>Historia de usuario:</b> 11
<b>Nombre:</b> Reporte de accesos.	
<b>Descripción:</b> Prueba para la funcionalidad de Reportar accesos.	
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	

<b>Respuesta del Sistema/Flujo Central:</b> <ul style="list-style-type: none"> <li>Al seleccionar el elemento "Acceso", el sistema muestra un panel que brinda la información sobre los accesos realizados ya sea con permisos concedidos o no a los diferentes locales .</li> </ul>
<b>Resultado Esperado:</b> Se han mostrado los diferentes accesos en el sistema.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

Fuente: elaboración propia

Tabla 20: Caso de prueba aceptación 13

Caso de Prueba Aceptación	
<b>Código:</b> 13	<b>Historia de usuario:</b> 17
<b>Nombre:</b> Insertar datos de local.	
<b>Descripción:</b> Prueba para la funcionalidad de insertar datos de local.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"> <li>La aplicación debe estar funcionando.</li> <li>El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b> <ul style="list-style-type: none"> <li>Al seleccionar la opción "Crear un nuevo local" el sistema muestra un panel que brinda la posibilidad de insertar los datos de una persona o de un local.</li> <li>Al seleccionar el botón "Crear" el sistema almacena los datos de un nuevo local en la base de datos.</li> </ul>	
<b>Resultado Esperado:</b> Se ha añadido el nuevo local en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 21: Caso de prueba aceptación 14

Caso de Prueba Aceptación	
<b>Código:</b> 14	<b>Historia de usuario:</b> 18
<b>Nombre:</b> Modificar datos de local.	
<b>Descripción:</b> Prueba para la funcionalidad de modificar datos de local.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"> <li>La aplicación debe estar funcionando.</li> <li>El administrador debe estar logueado.</li> </ul>	

<p><b>Respuesta del Sistema/Flujo Central:</b></p> <ul style="list-style-type: none"> <li>Al seleccionar la opción "editar" el sistema muestra un panel que brinda la posibilidad de editar los datos de un local.</li> </ul>
<p><b>Resultado Esperado:</b> Se han editado los datos del local en el sistema.</p>
<p><b>Evaluación de la Prueba:</b> Prueba satisfactoria.</p>

Fuente: elaboración propia

Tabla 22: Caso de prueba aceptación 15

Caso de Prueba Aceptación	
<b>Código:</b> 15	<b>Historia de usuario:</b> 19
<b>Nombre:</b> Listar local.	
<b>Descripción:</b> Prueba para la funcionalidad de listar local.	
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>La aplicación debe estar funcionando.</li> <li>El administrador debe estar logueado.</li> </ul>	
<p><b>Respuesta del Sistema/Flujo Central:</b></p> <ul style="list-style-type: none"> <li>Al seleccionar el elemento "Locales", el sistema muestra un panel con un listado de los datos de cada local.</li> </ul>	
<p><b>Resultado Esperado:</b> Se han mostrado los datos de los locales en el sistema.</p>	
<p><b>Evaluación de la Prueba:</b> Prueba satisfactoria.</p>	

Fuente: elaboración propia

Tabla 23: Caso de prueba aceptación 16

Caso de Prueba Aceptación	
<b>Código:</b> 16	<b>Historia de usuario:</b> 20
<b>Nombre:</b> Eliminar local.	
<b>Descripción:</b> Prueba para la funcionalidad de eliminar local.	
<p><b>Condiciones de Ejecución:</b></p> <ul style="list-style-type: none"> <li>La aplicación debe estar funcionando.</li> <li>El administrador debe estar logueado.</li> </ul>	
<p><b>Respuesta del Sistema/Flujo Central:</b></p> <ul style="list-style-type: none"> <li>Después de marcar la casilla, al seleccionar la opción "Eliminar" el sistema muestra un panel que brinda la posibilidad de eliminar a un local.</li> </ul>	

<b>Resultado Esperado:</b> Se ha eliminado el local en el sistema.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

Fuente: elaboración propia

Tabla 24: Caso de prueba aceptación 17

Caso de Prueba Aceptación	
<b>Código:</b> 17	<b>Historia de usuario:</b> 21
<b>Nombre:</b> Eliminar múltiples locales.	
<b>Descripción:</b> Prueba para la funcionalidad de eliminar múltiples locales.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después de marcar las casillas, al seleccionar la opción "Eliminar" el sistema muestra un panel que brinda la posibilidad de eliminar a varios locales.</li> </ul>	
<b>Resultado Esperado:</b> Se han eliminado a los locales en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 25: Caso de prueba aceptación 18

Caso de Prueba Aceptación	
<b>Código:</b> 18	<b>Historia de usuario:</b> 13
<b>Nombre:</b> Asignar grupo de acceso a persona.	
<b>Descripción:</b> Prueba para la funcionalidad de asignar grupo de acceso a persona.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después seleccionar la opción "Registrar tarjeta", el sistema muestra un panel que brinda la posibilidad de asignarle a una tarjeta el acceso a uno o varios locales.</li> </ul>	
<b>Resultado Esperado:</b> Se han asignado uno o varios locales a una persona en el sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 26: Caso de prueba aceptación 19

Caso de Prueba Aceptación	
<b>Código:</b> 19	<b>Historia de usuario:</b> 14
<b>Nombre:</b> Acceder a local.	
<b>Descripción:</b> Prueba para la funcionalidad de acceder a local.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después de pasar una tarjeta por el lector <i>RFID</i>, el sistema muestra una dirección URL que contiene el número de la tarjeta y el id de un local y se muestra un panel con el mensaje “true” o “false” que indica si se pudo acceder o no a un terminado local.</li> </ul>	
<b>Resultado Esperado:</b> Se pudo acceder o no a un determinado local.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 27: Caso de prueba aceptación 20

Caso de Prueba Aceptación	
<b>Código:</b> 20	<b>Historia de usuario:</b> 15
<b>Nombre:</b> Comprobar acceso a local.	
<b>Descripción:</b> Prueba para la funcionalidad de comprobar acceso a local.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después de pasar una tarjeta por el lector <i>RFID</i>, el servidor debe mandar una respuesta al cliente definiendo si se pudo acceder o no a un determinado local evidenciándose con el color del bombillo led, amarillo que sí y rojo que no.</li> </ul>	
<b>Resultado Esperado:</b> Se pudo comprobar el acceso a un determinado local.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 28: Caso de prueba aceptación 21

Caso de Prueba Aceptación	
<b>Código:</b> 21	<b>Historia de usuario:</b> 16
<b>Nombre:</b> Notificar.	
<b>Descripción:</b> Prueba para la funcionalidad de notificar.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• La aplicación debe estar funcionando.</li> <li>• El administrador debe estar logueado.</li> </ul>	
<b>Respuesta del Sistema/Flujo Central:</b>	
<ul style="list-style-type: none"> <li>• Después de pasar una tarjeta por el lector <i>RFID</i>, el servidor debe mandar una notificación al correo del administrador o de los directivos por cada área con el siguiente mensaje “Hola, se ha detectado un acceso sospechoso en el sistema” y los siguientes datos del acceso: tarjeta, persona, local y fecha.</li> </ul>	
<b>Resultado Esperado:</b> Se pudo enviar una notificación al correo del administrador o de los directivos.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Fuente: elaboración propia

### Resultados de las pruebas de caja negra

Fueron necesarias dos iteraciones para alcanzar un estado correcto del sistema implementado. En la primera iteración se detectaron las siguientes NC<sup>9</sup>:

- El número de solapín admitía otras letras que no fueran “E” y “T”.
- Cuando se seleccionaba la persona como “visitante” no se inhabilitaban los campos: usuario, área y foto.
- El “obtener datos de persona dado usuario”, se puede hacer siendo la persona como un “visitante”.
- En el reporte de accesos no se muestra el nombre de la persona que accedió a un determinado local.

En la segunda iteración fueron resueltas y no se detectaron nuevas NC.

<sup>9</sup> NC: no conformidades.

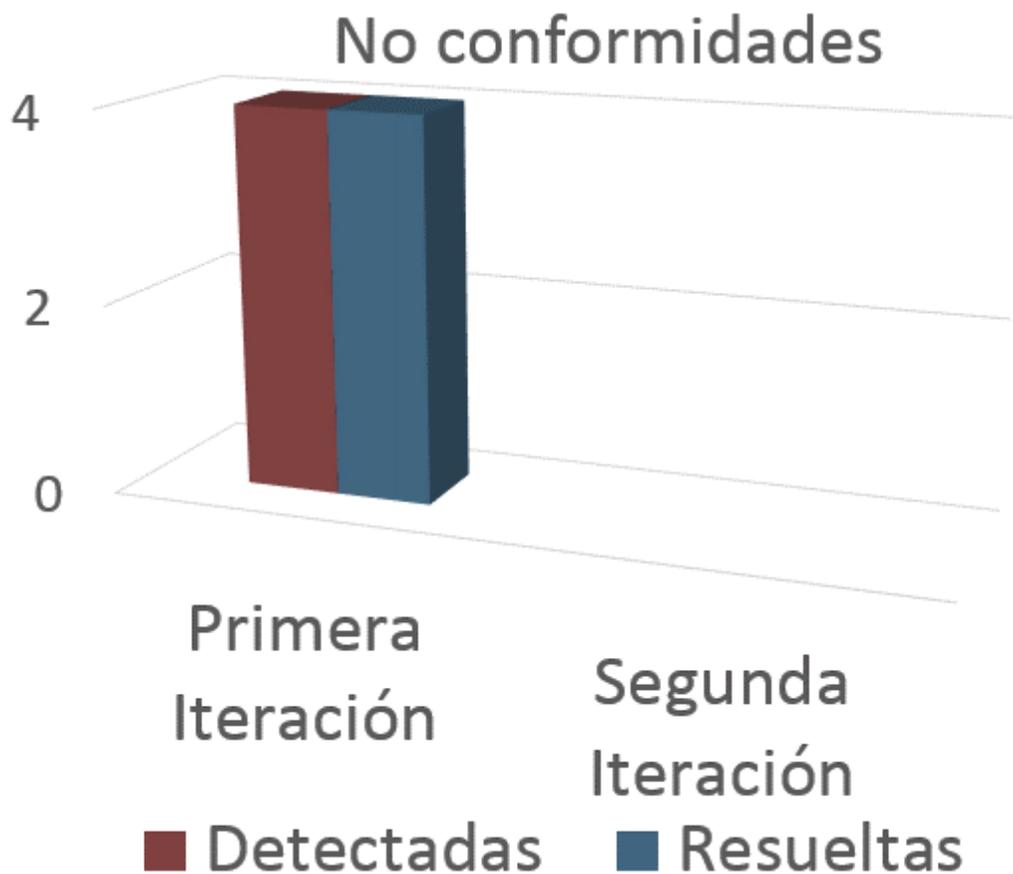


Figura 7: Resultados de las pruebas de caja negra  
Fuente: elaboración propia

### 3.4.2 Pruebas de Caja Blanca

Estas pruebas, también suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones o bloques; que han sido ejecutados (42).

Las principales técnicas de diseño de pruebas de caja blanca son:

- Pruebas de flujo de control.
- Pruebas de flujo de datos.
- Pruebas de bifurcación (*branch testing*).

- Pruebas de caminos básicos.

### **Pruebas de unidad**

Este tipo de prueba se centra en ejecutar cada módulo o unidad mínima a ser probada; lo cual provee un mejor modo de manejar la integración de las unidades en componentes mayores. Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido.

Para la presente investigación se utiliza este tipo de prueba mediante el uso de la herramienta PHPUnit<sup>10</sup>. Fueron necesarias tres iteraciones de pruebas hasta alcanzar que el sistema estuviera listo, en la primera iteración se detectaron las siguientes no conformidades (NC):

- EL usuario ingresa valores diferentes a los que la aplicación recibe en la tabla persona.
- Se almacenan tipos de datos de cadenas donde se debe almacenar enteros en la tabla tarjeta.
- Se accede a una posición en un arreglo la cual no existe en la clase TarjetaController.php.
- Existencia de variables con el mismo nombre, aunque sean diferentes en la clase Acceso.php.
- Uso incorrecto del punto y coma al definir una sentencia "if" en el método deleteAction() de la clase PersonaController.php.

En la segunda iteración fueron resueltas cinco NC y se detectaron otras dos:

- Mal uso del operador de asignación con el operador de comparación en el método accederAction() de la clase DefaultController de la entidad Acceso.
- No regresar un valor de retorno en todos los flujos del método editAction() de la clase TarjetaController.

En la tercera iteración de pruebas fueron resueltas las NC y no se detectaron nuevos errores. A continuación, se muestra un gráfico generado por dicha herramienta mostrando las no conformidades encontradas durante la realización de las pruebas al código.

---

<sup>10</sup> Entorno para realizar pruebas unitarias en el lenguaje de programación PHP.

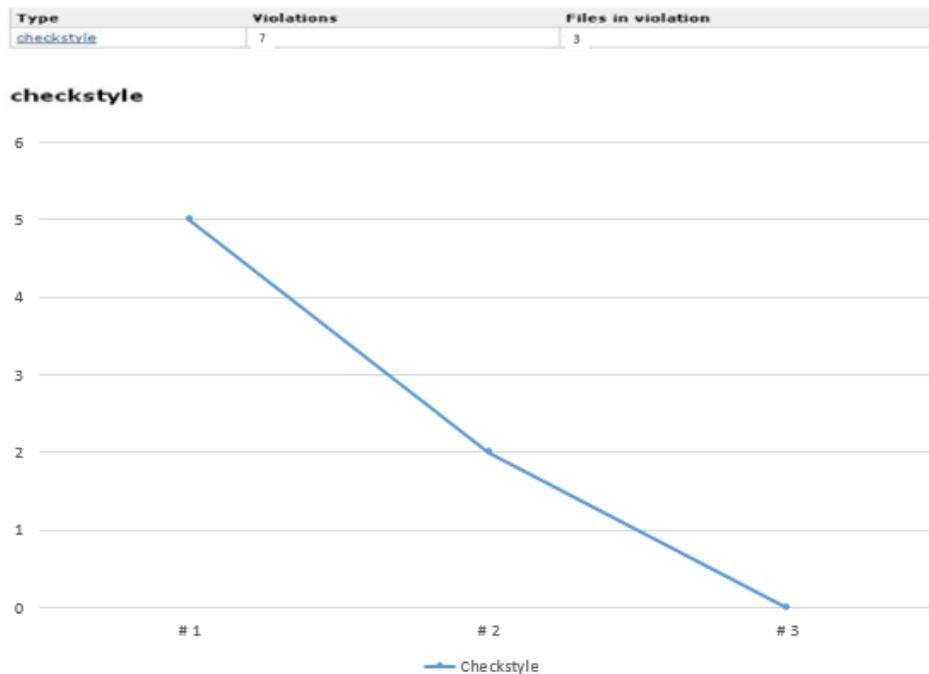


Figura 8: Resultados de las pruebas de unidad

Fuente: elaboración propia

### Conclusiones del capítulo

En el presente capítulo se realizó la elaboración del modelo de componentes que permitió representar en términos de componentes, los elementos descritos en el diseño. La elaboración del modelo de despliegue permitió representar la distribución física del sistema. Las pruebas permitieron encontrar errores que fueron solucionados en su totalidad.

## Conclusiones generales

Con el desarrollo de la presente investigación se arriba a las siguientes conclusiones:

1. El análisis de los principales aspectos y conceptos relacionados con los sistemas de control de acceso permitió obtener un mejor entendimiento de los mismos dentro del marco teórico de la investigación.
2. Después de analizar todas las tecnologías disponibles para el control de acceso; por su agilidad, confiabilidad y seguridad, la *RFID* es la más apropiada para la implementación del sistema a desarrollar.
3. Se desarrolló el sistema de control de acceso que permitió controlar el acceso a áreas restringidas y la entrada y salida a los locales en la UCI, el cual permitirá su despliegue sobre sistemas operativos libres.
4. El proceso de pruebas desarrollado, permitió identificar y resolver los errores en la implementación, garantizando el correcto funcionamiento del producto final, obteniéndose una aplicación que cumple con los requisitos definidos y que satisface las necesidades establecidas en los mismos.

## **Recomendaciones**

- Adaptar el sistema de control de acceso de manera que pueda ser utilizado no solo en la universidad, sino en cualquier centro que requiera controlar el acceso de personal.
- Vincular una cámara con el sistema de control de acceso para tomarle fotos a los visitantes.

## Referencias Bibliográficas

1. MUÑOZ, Emilio. *INNOVACIONES DISRUPTIVAS EN EL USO DE LAS TIC CON OBJETIVOS DE DESARROLLO SOCIAL* [online]. Madrid, 2015. Available from: [http://www.newwaysustainability.org/wp-content/uploads/2016/03/Informe-TICS-eptisa-14.03.2016\\_redu.pdf](http://www.newwaysustainability.org/wp-content/uploads/2016/03/Informe-TICS-eptisa-14.03.2016_redu.pdf)
2. SILVA, Rafaela. Tecnologías de control de acceso: factores importantes en la elección de soluciones | Anixter. [online]. 2016. [Accessed 5 March 2018]. Available from: [https://www.anixter.com/es\\_la/about-us/news-and-events/news/access-control-technologies-important-factors-when-choosing-a-solution.html](https://www.anixter.com/es_la/about-us/news-and-events/news/access-control-technologies-important-factors-when-choosing-a-solution.html)
3. RODRÍGUEZ SÁNCHEZ, Tamara. *Metodología de desarrollo para la Actividad productiva de la UCI*. 2015.
4. PALACIOS, Anabeni. Alarmas antirrobo para negocios ¿Cómo funcionan? *Distribuidor oficial Alarmas Tyco* [online]. 19 January 2017. [Accessed 11 April 2018]. Available from: <http://www.distribuidordealarmas.com/alarmas-antirrobo/Una manera muy común de proteger un negocio o comercio es a través de alarmas antirrobo>.
5. MICHAEL DAVIS and HANNIBAL. *Card Reader* [online]. 2017. Available from: [https://www.softconserv.com/downloads/manuals/SCS\\_How%20To/Card%20Readers.pdf](https://www.softconserv.com/downloads/manuals/SCS_How%20To/Card%20Readers.pdf)
6. DIPOLE. Tecnología RFID. [online]. 2017. [Accessed 14 March 2018]. Available from: <http://www.dipolerfid.es/es/tecnologia-RFID>
7. SEN, Jaydip. *ADVANCES IN SECURITY IN COMPUTING AND COMMUNICATIONS*. 2017.
8. SPACE. *System Assessment and Validation for Emergency Responders (SAVER)* [online]. 2015. Available from: [https://www.dhs.gov/sites/default/files/publications/ACT-HB\\_0915-508.pdf](https://www.dhs.gov/sites/default/files/publications/ACT-HB_0915-508.pdf)
9. ANZIL, Federico. *Concepto de Control*. . 2010.
10. What is access control? - Definition from WhatIs.com. [online]. 2017. Available from: <https://searchsecurity.techtarget.com/definition/access-control>
11. CEDEÑO MACÍAS, Cristian Eduardo and QUIROZ VERA, Miguel Angel. *Control de acceso por medio de cámaras y sensores en la empresa MELPROYECT de la ciudad Portoviejo* [online]. 2015. Available from: <http://repositorio.espm.edu.ec/bitstream/42000/336/1/TC82.pdf>
12. LÓPEZ PÉREZ, Nicolás. *TECNICAS DE BIOMETRIA BASADAS EN PATRONES FACIALES DEL SER HUMANO* [online]. UNIVERSIDAD TECNOLOGICA DE PEREIRA, 2012. Available from: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2738/0053682L864.pdf>
13. TEJADA MORALES, Diana Shirley. *Prototipo de Control de Acceso Peatonal al Campus de la Corporación Universitaria Lasallista* [online]. Caldas Antioquia, 2012. Available from: [http://repository.lasallista.edu.co/dspace/bitstream/10567/901/1/Prototipo\\_Control\\_Acceso\\_Corporacion\\_Universitaria\\_Lasallista.pdf](http://repository.lasallista.edu.co/dspace/bitstream/10567/901/1/Prototipo_Control_Acceso_Corporacion_Universitaria_Lasallista.pdf)
14. Bar Code Glossary of Terms. [online]. 2013. Available from: <http://www.adams1.com/barglos.html>

15. SANDOVAL SÁNCHEZ, Adela. *Propuesta de Diseño de Implementación del Sistema de Código de Barras en el Departamento de Registros Médicos y Servicios de Apoyo Diagnóstico en el Hospital San Juan de Dios* [online]. San José, Costa Rica, 2008. Available from: <http://www.binasss.sa.cr/libros/0648.pdf>
16. ALVARADO SÁNCHEZ, Jorge Alberto. *Sistema de Control de Acceso con RFID* [online]. México, D.F, 2008. Available from: <https://www.cs.cinvestav.mx/Estudiantes/TesisGraduados/2008/tesisJorgeAlvarado.pdf>
17. GUZMÁN MARTÍNEZ, Ricardo Gabriel. *Implantación de Código de Barras en un almacén de electrodomésticos* [online]. México, D.F : UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, 2008. Available from: [http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2388/Tesis Ricardo Gabriel Guzman Mtz-Codigo de barras.pdf?sequence=1](http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2388/TesisRicardoGabrielGuzmanMtz-Codigo%20de%20barras.pdf?sequence=1)
18. RFID POINT. Control de Activos con Tecnología RFID en México. *RFID POINT* [online]. 2012. [Accessed 31 October 2017]. Available from: [http://www.rfidpoint.com/control-de-activos-con-tecnologia-rfid-en-mejico/Los bancos y otras empresas de la ciudad de México están utilizando etiquetas pasivas UHF EPC para hacer un seguimiento de su mobiliario y están probando un sistema que usa etiquetas activas para detectar cuando se están utilizando áreas específicas de las oficinas. La división...](http://www.rfidpoint.com/control-de-activos-con-tecnologia-rfid-en-mejico/Los%20bancos%20y%20otras%20empresas%20de%20la%20ciudad%20de%20M%C3%A9xico%20est%C3%A1n%20utilizando%20etiquetas%20pasivas%20UHF%20EPC%20para%20hacer%20un%20seguimiento%20de%20su%20mobiliario%20y%20est%C3%A1n%20probando%20un%20sistema%20que%20usa%20etiquetas%20activas%20para%20detectar%20cuando%20se%20est%C3%A1n%20utilizando%20%C3%A1reas%20espec%C3%ADficas%20de%20las%20oficinas.%20La%20divisi%C3%B3n...)
19. RFID POINT. RFID POINT - La comunidad de RFID en Latinoamérica. [online]. 2016. [Accessed 31 October 2017]. Available from: <http://www.rfidpoint.com/>
20. WALKER, John. RFID ayuda en el traslado de carga congelada de H&M Bay. [online]. 2013. Available from: <http://www.rfidpoint.com/casos-de-exito/solucion-rfid-de-motorola-ayuda-en-el-traslado-de-carga-congelada-de-hm-bay/>
21. GUEVARA, Yurisander. Balitas automatizadas. [online]. 2016. [Accessed 10 November 2017]. Available from: [http://www.juventudrebelde.cu/suplementos/informatica/2016-04-06/balitas-automatizadasCuba desarrolla un novedoso sistema informático para mejorar el control y la productividad con los cilindros de gas licuado](http://www.juventudrebelde.cu/suplementos/informatica/2016-04-06/balitas-automatizadasCuba%20desarrolla%20un%20novedoso%20sistema%20inform%C3%A1tico%20para%20mejorar%20el%20control%20y%20la%20productividad%20con%20los%20cilindros%20de%20gas%20licuado)
22. VIRTUALWARE. Aplicación y mesa multitáctil, Palacio del II Cabo en Cuba | Virtualware. [online]. 2016. [Accessed 10 November 2017]. Available from: <http://virtualwaregroup.com/es/portfolio/aplicacion-y-mesa-multitactil-palacio-del-ii-cabo-cuba>
23. MAIDA and ESTEBAN GABRIEL. *Metodologías de desarrollo de software* [online]. Argentina, Buenos Aires, 2015. Available from: <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>
24. MONTOYA MALDONADO, Lourdes, OSCCO HUANGAL, Crithian Junior and MEGO JULCA, Lucio. Lenguaje de modelado unificado uml. [online]. 2016. Available from: <https://es.slideshare.net/turlahackers/lenguaje-de-modelado-unificado-uml>
25. LÓPEZ ORTEGA, Daniel and SANTA VILLA, Jessica Andrea. *ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS CASE: STARUML, POSEIDON FOR UML Y ENTERPRISE ARCHITECT, PARA EL MODELAMIENTO DE DIAGRAMAS UML* [online]. 2012. Available from: <http://recursosbiblioteca.utp.edu.co/tesis/textoyanexos/0053L864e.pdf>
26. ORLANDO. Módulo Lector RFID-RC522 RF con Arduino UNO R3 SPI. [online]. 2014. [Accessed 17 September 2017]. Available from: <https://hetpro-store.com/TUTORIALES/modulo-lector-rfid-rc522-rf-con-arduino/>

27. RC522 RFID Module 13.56MHz | RC522-READER |. [online]. 2014. [Accessed 9 November 2017]. Available from: <http://www.hobbytronics.co.uk/mfrc522-reader>
28. GARCÍA BARRAGÁN, Marcelino. OUNO Arduino UNO genérico. *HeTPro* [online]. 2017. [Accessed 9 November 2017]. Available from: <https://hetpro-store.com/ouno-arduino-uno-generico/OUNO> Arduino UNO genérico es una tarjeta de desarrollo basada en el microcontrolador ATmega328. Tiene 14 entradas y salidas digitales.
29. ESTRADA MARMOLEJO, Rubén. Qué es Arduino. *HeTPro* [online]. 2017. [Accessed 3 December 2017]. Available from: <https://hetpro-store.com/que-es-arduino/Qué es Arduino>. Arduino es un proyecto electrónico que involucra Hardware y Software libre mediante una interfaz simple.
30. CRESPO, Jose Enrique. Ethernet Shield | Aprendiendo Arduino. [online]. 2016. [Accessed 21 November 2017]. Available from: <https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>
31. TORRES HERNÁNDEZ, Miguel. *Symfony Framework. Desarrollo Rápido de Aplicaciones Web*. 2015.
32. MIFSUF TALÓN, Elvira. *Apache: servidor web* [online]. 2013. Available from: <http://descargas.pntic.mec.es/mentor/visitas/Apache.pdf>
33. PALOMO DUARTE, Manuel. *Programación en PHP a través de ejemplos* [online]. Sevilla España, 2014. Available from: [http://servicio.uca.es/softwarelibre/publicaciones/apuntes\\_php](http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php)
34. BENJUMEA, Vicente and ROLDÁN, Manuel. *Fundamentos de programación con el lenguaje de programación c++* [online]. 2017. Available from: [http://www.lcc.uma.es/~vicente/docencia/cppdoc/programacion\\_cxx.pdf](http://www.lcc.uma.es/~vicente/docencia/cppdoc/programacion_cxx.pdf)
35. Tutor de C++. [online]. 2014. [Accessed 3 December 2017]. Available from: [http://decsai.ugr.es/~jfv/ed1/c++/cdrom3/TIC-CD/web/portada/faqs/faqs1\\_3.htm](http://decsai.ugr.es/~jfv/ed1/c++/cdrom3/TIC-CD/web/portada/faqs/faqs1_3.htm)
36. FERGACIAC. Entorno de Desarrollo Integrado (IDE). [online]. 2013. [Accessed 20 January 2018]. Available from: <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>
37. GONZÁLEZ PALOMARES, Rafael José. *Plataforma de valoración y venta de consolas, juegos y accesorios: versión web*. 2016.
38. Arduino - Software. [online]. 2017. [Accessed 21 November 2017]. Available from: <https://www.arduino.cc/en/Main/Software>
39. URUELA, Juan. Los gestores de bases de datos más usados. [online]. 2016. Available from: <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>
40. CONTRIBUTORS, phpMyAdmin. phpMyAdmin. *phpMyAdmin* [online]. 2018. [Accessed 13 March 2018]. Available from: <https://www.phpmyadmin.net/>
41. cerraduras electronicas rfid. [online]. Available from: <https://www.google.com/cu/search?q=cerraduras+electronicas+rfid&tbm=isch&tbo=u&source=univ&sa=X&ved=0ahUKEwjX-PG7yNbbAhUs6YMKHcsMCSMQsAQIYg&biw=1366&bih=645>

42. PRESSMAN, Roger S. *Ingeniería del Software. Un Enfoque Práctico*. 2010.
43. SOMMERVILLE, (primero). *Ingeniería de software*. 9na edición. 2005.
44. HERNANDEZ, Uriel. MVC (Model, View, Controller) explicado. [online]. 2015. [Accessed 27 February 2018]. Available from: <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
45. CASAS DE LA TORRE, Fernando Alfonso. Patrones GRASP. [online]. Software. 2017. [Accessed 10 March 2018]. Available from: [https://es.slideshare.net/Indiana\\_1969/patrones-grasp-76283581](https://es.slideshare.net/Indiana_1969/patrones-grasp-76283581) Presentación de los patrones GRASP enfocados al análisis y diseño de sistemas
46. ARIAS CALLEJA, Manuel. *Carmen. Estándares de codificación* [online]. [no date]. Available from: <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>
47. SUAZO LA ROSA, Cristian. Modelo de Implementación Diagramas de Componentes y Despliegue. *Scribd* [online]. 2013. [Accessed 3 April 2018]. Available from: <https://es.scribd.com/document/100478725/Modelo-de-Implementacion-Diagramas-de-Componentes-y-Despliegue> Scribd es el sitio social de lectura y editoriales más grande del mundo.
48. PRESSMAN, Roger S. *Ingeniería del Software (Un enfoque práctico)*. 7th edition. New York, EEUU, 2010.
49. Placas Arduino/Genuino: guía de compras con modelos compatibles y kits de iniciación. [online]. 2016. Available from: <https://www.xataka.com/makers/empezar-con-arduino-genuino-como-elegir-la-placa-modelos-compatibles-y-kits-de-iniciacion>
50. Demarkt Placa Arduino con Ethernet Shield W5100 - Servidor Web ,Domotica: Amazon.es: Informática. [online]. 2016. Available from: <https://www.amazon.es/Demarkt-Arduino-Ethernet-Shield-W5100/dp/B01HMIU9VM>

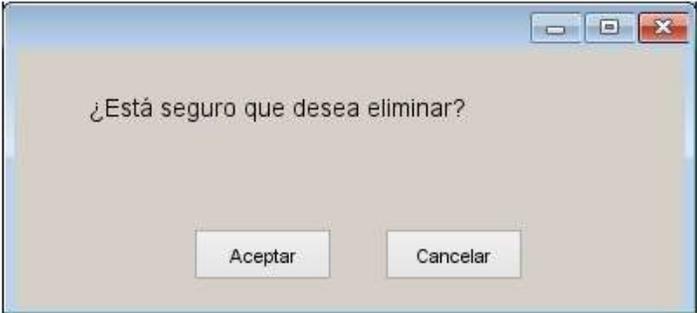
## Bibliografía

1. ¿Qué es la autorización y el control de acceso? [online]. [Accessed 11 April 2018]. Available from: <https://www.icann.org/news/blog/que-es-la-autorizacion-y-el-control-de-acceso>
2. PALACIOS, Anabeni. Alarmas antirrobo para negocios ¿Cómo funcionan? Distribuidor oficial Alarmas Tyco [online]. 19 January 2017. [Accessed 11 April 2018]. Available from: <http://www.distribuidordealarmas.com/alarmas-antirrobo/Una-manera-muy-comun-de-proteger-un-negocio-o-comercio-es-a-traves-de-alarmas-antirrobo>.
3. Lectura de tarjetas RFID con Arduino y lector MIFARE RC522. [online]. [Accessed 13 February 2018]. Available from: <https://www.luisllamas.es/arduino-rfid-mifare-rc522/>
4. Arduino + ethernet Shield + Rfid + Mysql +Nodejs - - AYUDA. [online]. [Accessed 27 March 2018]. Available from: <http://forum.arduino.cc/index.php?topic=275269.0>
5. GÓMEZ, Enrique. De MySQL a Arduino, leer datos de la base de datos. RincónIngenieril [online]. 17 January 2015. [Accessed 27 March 2018]. Available from: <https://www.rinconingenieril.es/de-mysql-a-arduino/>
6. Historias de usuario, una forma natural de análisis funcional. [online]. [Accessed 19 February 2018]. Available from: <https://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>
7. BILLY REYNOSO, Carlos. Introducción a la Arquitectura de Software [online]. Universidad de Buenos Aires, 2004. Available from: [http://artemisa.unicauca.edu.co/~cardila/IS\\_05\\_\\_Lectura\\_\\_Conceptos\\_Arquitectura\\_SW.pdf](http://artemisa.unicauca.edu.co/~cardila/IS_05__Lectura__Conceptos_Arquitectura_SW.pdf)
8. Symfony 1.0, la guía definitiva. [online]. [Accessed 15 February 2018]. Available from: [http://librosweb.es/libro/symfony\\_1\\_0/ HTTP](http://librosweb.es/libro/symfony_1_0/ HTTP)
9. Welcome! - The Apache Server Project. [online]. 2017. [Accessed 20 January 2018]. Available from: <https://httpd.apache.org/>
10. Test all the Things! Types and Examples of Software Tests. [online]. Available from: <https://www.netguru.co/blog/software-testing>

## Anexos

### Anexo 1: HU del RF3

Tabla 29: HU del RF3

Número: HU 3		Nombre del requisito: Eliminar persona	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"><li>• Problemas de fallos de conexión.</li><li>• Planificación incorrecta.</li></ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir eliminar el área, la cual será el centro al que pertenece la persona, el nombre de la persona, los apellidos, el solapín, la foto y el usuario serán tomados del directorio UCI, el cargo será estudiante, trabajador y visitante y el sexo de una persona.			
<b>Observaciones:</b> <ul style="list-style-type: none"><li>•</li></ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			
			

Fuente: elaboración propia

### Anexo 2: HU del RF4

Tabla 30: HU del RF4

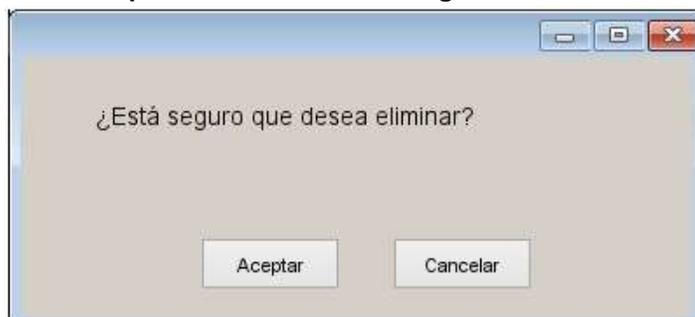
Número: HU 4		Nombre del requisito: Eliminar múltiples personas	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"><li>• Problemas de fallos de conexión.</li><li>• Planificación incorrecta.</li></ul>		<b>Tiempo Real:</b> 2 días	

**Descripción:**

El sistema deberá permitir eliminar a varias personas a la vez después de ser seleccionadas.

**Observaciones:**

- 

**Prototipo elemental de interfaz gráfica de usuario:**

Fuente: elaboración propia

## Anexo 3: HU del RF5

Tabla 31: HU del RF5

<b>Número:</b> HU 5		<b>Nombre del requisito:</b> Listar persona	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir listar las personas almacenadas en el sistema y mostrar los datos de dichas personas, como lo son: el solapín, carnet de identidad en caso de los visitantes, área, el nombre, los apellidos, la foto, el usuario, cargo y el sexo.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>• Si la persona es un visitante solo serán mostrados los datos carnet de identidad(CI), nombre, apellidos, sexo y cargo.</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			

**Listado de personas**

<input type="checkbox"/>	Foto	Usuario	Nombre	Apellidos	Solapín/CI	Sexo	Cargo	Area	Acciones
<input type="checkbox"/>		yelva	Yavel	Silva Gomez	E118024	masculino	Estudiante	Facultad 4	<input type="button" value="Editar"/>
<input type="checkbox"/>		ggonzalez	Gustavo Gonzalez Diaz	Gonzalez Diaz	E123456	masculino	Estudiante	Facultad 4	<input type="button" value="Editar"/>

Fuente: elaboración propia

## Anexo 4: HU del RF7

Tabla 32: HU del RF7

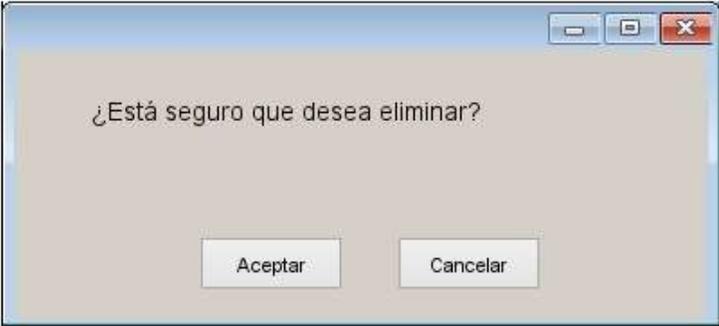
<b>Número:</b> HU 7		<b>Nombre del requisito:</b> Modificar estado de tarjeta	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 días	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir modificar el estado de una tarjeta a habilitada o no.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>• Si la tarjeta está habilitada es porque ya está asignada a alguna persona.</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			
			

Fuente: elaboración propia

## Anexo 5: HU del RF8

Tabla 33: HU del RF8

<b>Número:</b> HU 8		<b>Nombre del requisito:</b> Eliminar tarjeta	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	

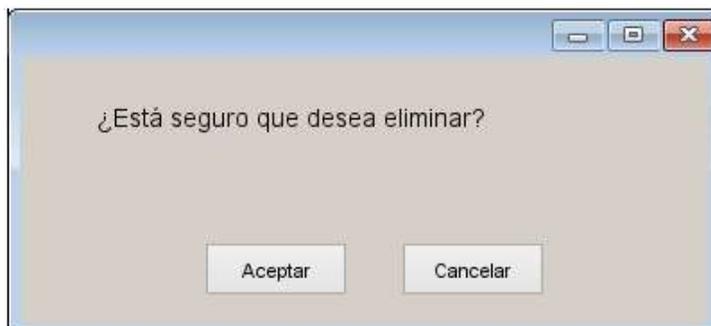
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>	<b>Tiempo Real:</b> 2 días
<b>Descripción:</b> El sistema deberá permitir eliminar el número de una tarjeta del sistema.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>•</li> </ul>	
<b>Prototipo elemental de interfaz gráfica de usuario:</b>	
	

Fuente: elaboración propia

## Anexo 6: HU del RF9

Tabla 34: HU del RF9

<b>Número:</b> HU 9		<b>Nombre del requisito:</b> Eliminar múltiples tarjetas	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir eliminar múltiples tarjetas del sistema después de ser seleccionadas.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>•</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			



Fuente: elaboración propia

## Anexo 7: HU del RF10

Tabla 35: HU del RF10

Número: HU 10		Nombre del requisito: Listar tarjeta	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b>			
El sistema deberá permitir mostrar todas las tarjetas existentes en el sistema, ya sea habilitada o no.			
<b>Observaciones:</b>			
<ul style="list-style-type: none"> <li>• Cuando la tarjeta está deshabilitada, no se podrá acceder a ningún local.</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			

Fuente: elaboración propia

## Anexo 8: HU del RF11

Tabla 36: HU del RF11

<b>Número:</b> HU 11		<b>Nombre del requisito:</b> Reportar accesos	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b>			
El sistema deberá permitir exportar un reporte de accesos en formato pdf sobre los datos de las personas que accedieron a un local, los cuales serán el nombre de la persona, el número de la tarjeta, además de la fecha y hora en que se hizo el acceso y muestra si es entrada o salida a dicho local. Este reporte será diario.			
<b>Observaciones:</b>			
<ul style="list-style-type: none"> <li>• El formato de la fecha será dd/mm/aa y el de la hora será hh/mm/ss y de 24 horas.</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			
			

Fuente: elaboración propia

## Anexo 9: HU del RF12

Tabla 37: HU del RF12

<b>Número:</b> HU 12		<b>Nombre del requisito:</b> Obtener datos de persona dado usuario	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b>			

El sistema deberá permitir obtener de las personas al ser creadas el área, la cual será el centro al que pertenece la persona, el nombre de la persona, los apellidos, el solapín y la foto dado el usuario, estos datos serán tomados del directorio UCI.

**Observaciones:**

- Todos estos datos obtenidos serán para las personas que sean estudiantes y trabajadores, pues los visitantes no tienen usuario.

**Prototipo elemental de interfaz gráfica de usuario:**

Fuente: elaboración propia

## Anexo 10: HU del RF13

Tabla 38: HU del RF13

<b>Número:</b> HU 13		<b>Nombre del requisito:</b> Asignar grupo de acceso a persona	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir asignar un grupo de acceso a cada persona, o sea, cada persona en el sistema tendrá acceso a uno o varios locales.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>•</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			

Fuente: elaboración propia

## Anexo 11: HU del RF14

Tabla 39: HU del RF14

<b>Número:</b> HU 14		<b>Nombre del requisito:</b> Acceder a local	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir que una persona pueda acceder a uno o varios locales.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>• Si la tarjeta no está registrada y no está habilitada, la persona no podrá acceder a ningún local.</li> </ul>			

### Prototipo elemental de interfaz gráfica de usuario:

Fuente: elaboración propia

## Anexo 12: HU del RF15

Tabla 40: HU del RF15

Número: HU 15		Nombre del requisito: Comprobar acceso a local	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"><li>• Problemas de fallos de conexión.</li><li>• Planificación incorrecta.</li></ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir comprobar si se accedió o no a un determinado local.			
<b>Observaciones:</b> <ul style="list-style-type: none"><li>• Si se encienden los bombillos leds del arduino es que el sistema envió la respuesta de que se accedió o no a un determinado local .</li></ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			

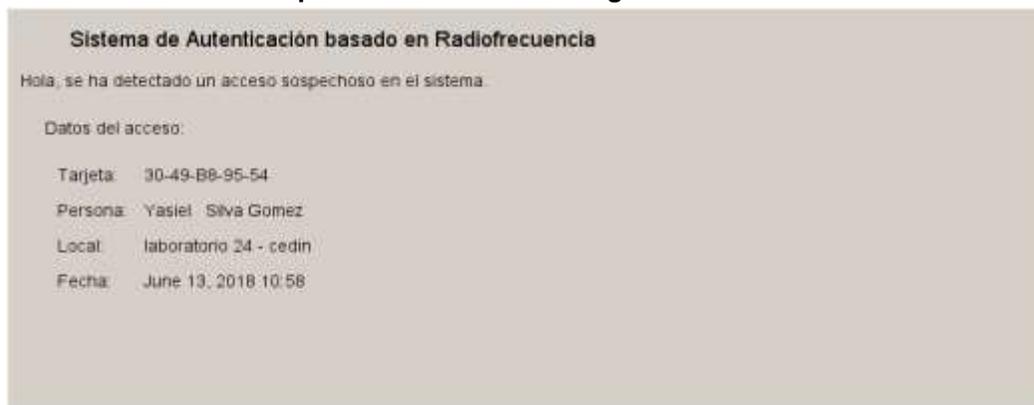
Fuente: elaboración propia

## Anexo 13: HU del RF16

Tabla 41: HU del RF16

Número: HU 16		Nombre del requisito: Notificar	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"><li>• Problemas de fallos de conexión.</li><li>• Planificación incorrecta.</li></ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir enviar un correo a la administración sobre el id, solapín, área, el nombre, los apellidos, la foto, el usuario, cargo y el sexo de la persona que intentó acceder a un local en horario no laborable, además de la fecha y hora. Este reporte se hará diario, si nadie intenta acceder, el sistema no hará ningún reporte.			
<b>Observaciones:</b> <ul style="list-style-type: none"><li>• Si la persona que intentó acceder no está en la base de datos del sistema, se enviará un correo al administrador del sistema informando el intento de acceso, o si se intenta acceder con una tarjeta que esté deshabilitada en el sistema, se mostrarán los datos de su antiguo propietario.<ul style="list-style-type: none"><li>• El horario laboral que está establecido es de 8:00 am a 17:00 pm.</li></ul></li></ul>			

**Prototipo elemental de interfaz gráfica de usuario:**



Fuente: elaboración propia

**Anexo 14: HU del RF17**

Tabla 42: HU del RF17

<b>Número:</b> HU 17	<b>Nombre del requisito:</b> Insertar datos de local	
<b>Programador:</b> Yasiel Silva Gómez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>	<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir insertar el nombre y responsable del local, además del área a la cual pertenece dicho local.		
<b>Observaciones:</b> •		

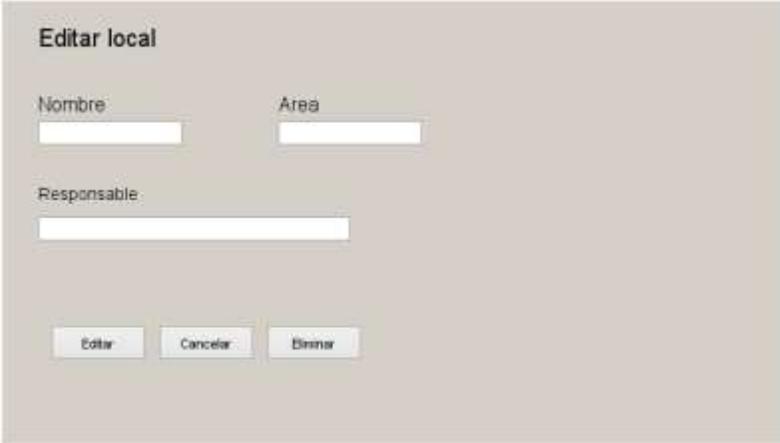
**Prototipo elemental de interfaz gráfica de usuario:**



Fuente: elaboración propia

## Anexo 15: HU del RF18

Tabla 43: HU del RF18

Número: HU 18		Nombre del requisito: Modificar datos de local	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir modificar el nombre y responsable del local, además del área a la cual pertenece dicho local.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>•</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			
			

Fuente: elaboración propia

## Anexo 16: HU del RF19

Tabla 44: HU del RF19

Número: HU 19		Nombre del requisito: Listar local	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> </ul>		<b>Tiempo Real:</b> 2 días	

- Planificación incorrecta.

**Descripción:**

El sistema deberá permitir listar los locales almacenados en la base de datos del sistema y mostrar los datos de dichos locales, como lo son el nombre y responsable del local, además del área a la cual pertenece dicho local.

**Observaciones:**

- 

**Prototipo elemental de interfaz gráfica de usuario:**

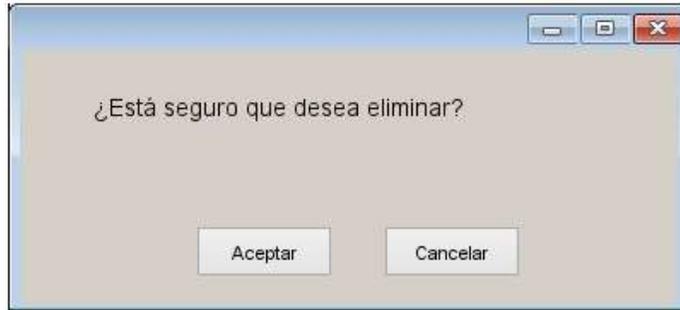


Fuente: elaboración propia

## Anexo 17: HU del RF20

Tabla 45: HU del RF20

Número: HU 20		Nombre del requisito: Eliminar local	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir eliminar el local del sistema con sus respectivos datos, los cuales son: el nombre y responsable del local, además del área a la cual pertenece dicho local.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>•</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			



Fuente: elaboración propia

## Anexo 18: HU del RF21

Tabla 46: HU del RF21

<b>Número:</b> HU 21		<b>Nombre del requisito:</b> Eliminar múltiples locales	
<b>Programador:</b> Yasiel Silva Gómez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Problemas de fallos de conexión.</li> <li>• Planificación incorrecta.</li> </ul>		<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> El sistema deberá permitir eliminar múltiples locales después de ser seleccionados.			
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>•</li> </ul>			
<b>Prototipo elemental de interfaz gráfica de usuario:</b>			

Fuente: elaboración propia

**Anexo 19: Diagrama de componentes correspondiente al caso de uso gestionar tarjeta y al requisito funcional asignar grupo de acceso a persona.**

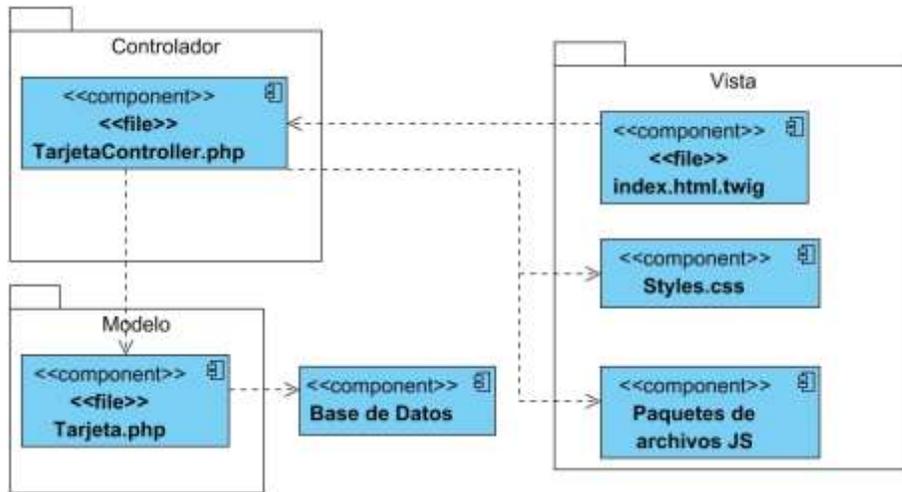


Figura 9: Diagrama de componentes gestionar tarjeta  
Fuente: elaboración propia

**Anexo 20: Diagrama de componentes correspondiente al caso de uso gestionar local.**

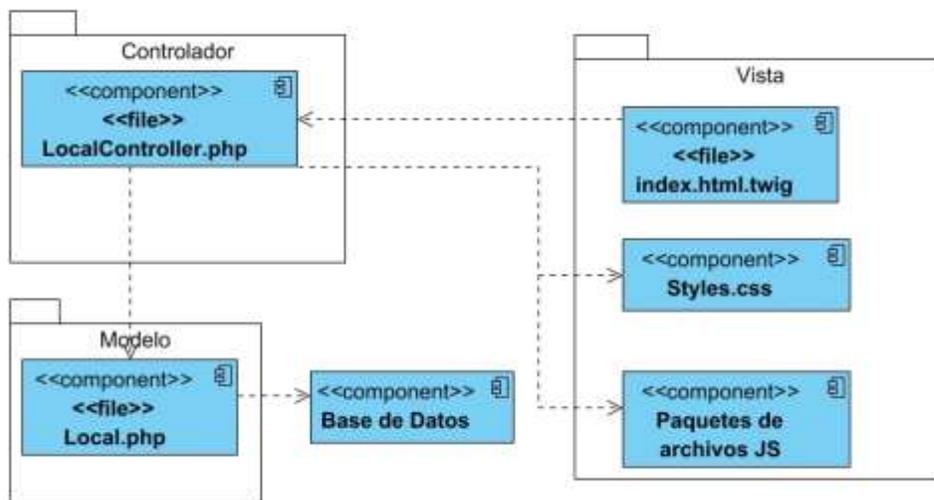


Figura 10: Diagrama de componentes gestionar local  
Fuente: elaboración propia

**Anexo 21: Diagrama de componentes correspondiente al requisito funcional obtener datos de persona dado usuario.**

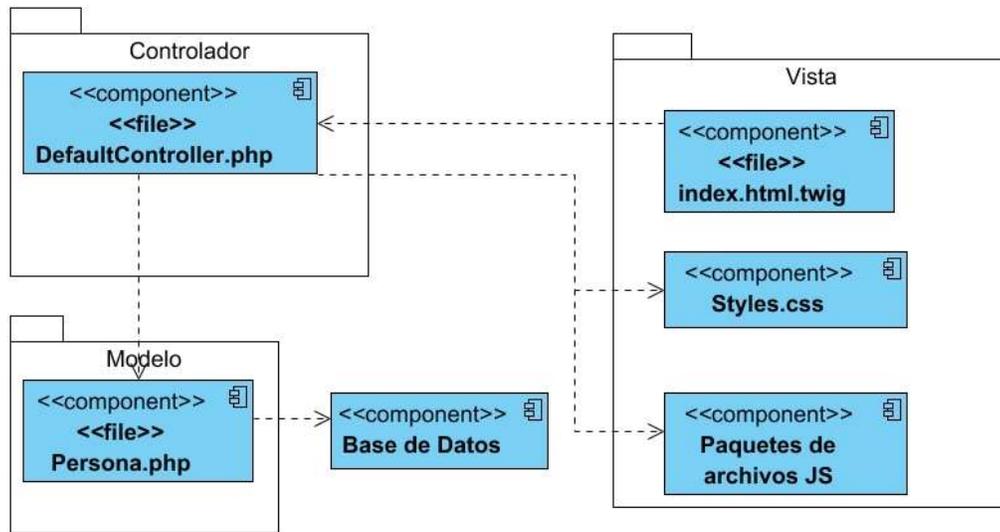


Figura 11: Diagrama de componentes obtener datos de persona dado usuario

Fuente: elaboración propia

## Anexo 22: Diagrama de componentes correspondiente a los requisitos funcionales acceder a local, notificar y comprobar acceso a local.

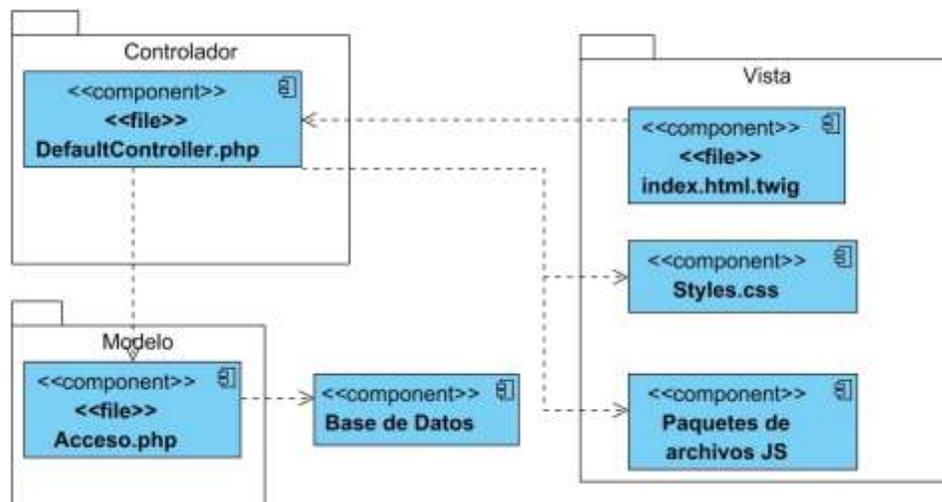


Figura 12: Diagrama de componentes acceder a local, notificar y comprobar acceso a local

Fuente: elaboración propia