

**Universidad de las Ciencias Informáticas**

**Facultad 4**



**Título:** Subsistema de control de acceso del sistema de gestión de alimentación de la Universidad de las Ciencias informáticas.

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

**Autor(es):** Roberto Rodríguez Caballero

**Tutor(es):** Ing. Julio César Espronceda Pérez

Ing. Yidier Romero Zaldívar

### Frase

“Un poco más de persistencia, un poco más de esfuerzo, y lo que parecía irremediadamente un fracaso puede convertirse en un éxito glorioso”.

Elbert Hubbard

## Declaración de autoría

Declaro ser autor de la presente tesis que tiene por título: Subsistema de control de acceso del sistema de gestión de alimentación y recomendación a las consultorías de las tecnologías de la información. Reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los 29 días del mes de junio del año 2018.

Roberto Rodríguez Caballero

Firma del Autor

Yidier Romero Zaldívar

Firma del Tutor

Julio Cesar Espronceda Pérez

Firma del Tutor

## Datos de contacto

*Ing. Yidier Romero Zaldívar.* Graduado de Ingeniero en Ciencias Informáticas en 2006. Actualmente pertenece al Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE). Tiene categoría de docente de Asistente y cuenta con 11 años de trabajo. Ha investigado en la disciplina de Inteligencia Artificial y ha impartido clases de pregrado en asignaturas homónimas. Investiga el tema de los sistemas distribuidos y arquitectura orientada a servicios desde el 2009.

*Ing. Julio César Espronceda Pérez.* Graduado de Ingeniero en Ciencias Informáticas en 2013. Actualmente se desempeña como Especialista Superior en la Dirección de Extensión Universitaria de la Universidad de las Ciencias Informáticas (UCI) e imparte clases, como profesor Asistente, en la disciplina de Ingeniería y Gestión de Software. Miembro de la Sociedad Cubana de Matemática y Computación (SCMC) y la Unión de Informáticos de Cuba (UIC).

## Agradecimientos

Primeramente, agradezco a la Universidad de la Ciencias informáticas por haberme aceptado ser parte de ella y abierto las puertas de su seno científico para poder estudiar mi carrera, así como también a los diferentes docentes que brindaron sus conocimientos y su apoyo para seguir adelante cada día.

Agradezco también a mis tutores de tesis el Ing. Julio Cesar Espronceda Pérez y el Ing. Yidier Romero Zaldívar por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento, así como también haberme tenido toda la paciencia del mundo para guiarme durante todo el desarrollo de la tesis.

Y para finalizar, también agradezco a todos mis amigos que han estado conmigo durante mis años de universidad por su amistad y compañerismo que han aportado un alto porcentaje a mis ganas de seguir adelante en mi carrera profesional.

## Dedicatoria

Dedicado a mi familia y amigos.

## Resumen

En todo el mundo, la informatización de los procesos de una entidad ha demostrado contribuir a lograr los objetivos de la misma, así como ganar en eficiencia y eficacia en el desarrollo de sus actividades. Cuba no está ajena a esta nueva tendencia, donde juega un papel muy importante la Universidad de las Ciencias Informáticas (UCI). La gestión de los servicios alimenticios es uno de los procesos que se han logrado informatizar. La UCI cuenta con una solución informática que gestiona sus servicios alimenticios. El subsistema de control de acceso del personal como parte del sistema de gestión de servicios alimenticios necesita ser migrado a software libre, así como actualizar el mismo dado las nuevas políticas del negocio que se están implementado en la casa de altos estudios. Por lo que en este trabajo de diploma se desarrollará un subsistema de control de acceso que responda a la política de software libre de la universidad y a las nuevas reglas de negocio en la misma. Como resultado se obtuvo la aplicación “Subsistema de control de acceso” contribuyendo a que no se formen atascos en las filas de los comedores y en el manejo de las interrupciones de la red además de permitir el monitoreo de los comensales que acceden por los puntos de acceso.

**PALABRAS CLAVE:** Acceso, Alimenticios , Gestión, Subsistema.

## SUMMARY

Throughout the world, the computerization of the processes of an entity has proven to contribute to achieving the objectives of the same as well as to gain in efficiency and effectiveness in the development of its activities. Cuba is not unaware of this new trend, where it plays an very important role the University of Computer Science (UCI). The management of food services is one of the processes that have been computerized. The University of Computer Science has a computer solution that manages their food services. The subsystem of Personnel access control as part of the food service management system needs to be migrated to free software, as well as updating it given the new business policies that are being implemented in the high school house. So this thesis will develop an access control subsystem that responds to the free software policy of the university and the new business rules in it. As a result, the application “Access Control Subsystem” was obtained, helping to prevent bottle necks from forming in the rows of the dining rooms and in the management of network interruptions.

**KEYWORDS:** Access, Food, Management, Subsystem.

## Tabla de contenidos

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTOS DE LA INVESTIGACIÓN, METODOLOGÍAS Y TECNOLOGÍAS EMPLEADAS.</b> .....	<b>6</b>
Introducción .....	6
1.1 Sistema de alimentación UCI. ....	6
1.2 Sistema de control de acceso basado en credenciales. ....	8
1.3 Componentes del sistema de control de acceso. ....	10
1.4 Proceso de control de acceso. ....	11
1.5 Descripción de los componentes del proceso de control de acceso. ....	12
1.6 Implicaciones de las Tendencias Recientes en la Arquitectura de Sistemas. ....	16
1.7 Sistemas de control de acceso estudiados a nivel internacional. ....	17
1.8 Sistemas de control de acceso estudiados desarrollados a nivel nacional. ....	19
1.9 Tecnologías, metodología, lenguaje y herramientas. ....	21
1.10 Metodologías de desarrollo de software .....	25
1.11 Herramientas de desarrollo. ....	28
1.12 Herramienta de modelado. ....	28
Conclusión parcial. ....	29
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.</b> .....	<b>30</b>
Introducción. ....	30
2.1 Modelo de dominio. ....	30
2.2 Propuesta de solución. ....	32
2.3 Requisitos. ....	33
2.3.1 Requisitos funcionales. ....	34
2.3.2 Requisitos no funcionales. ....	35
2.4 Historia de Usuario. ....	36
2.5 Arquitectura .....	40
2.5.1 Patrones de diseño y estilo de arquitectura. ....	41
<b>CAPÍTULO 3: EVALUACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA</b> .....	<b>43</b>
3.1 Tareas de ingeniería. ....	43
3.2 Estándares de codificación. ....	45
3.2 Diagrama de componente. ....	46
3.3 Diagrama de despliegue. ....	46
3.4 Pruebas .....	47
3.4.1 Pruebas de caja negra .....	48
3.5 Conclusiones Parciales. ....	55
<b>CONCLUSIONES</b> .....	<b>56</b>



<b>RECOMENDACIONES.....</b>	<b>57</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>58</b>
<b>ANEXOS.....</b>	<b>60</b>
<b>Anexo 1: Historias de usuario .....</b>	<b>60</b>
<b>Anexo 2: Tareas de ingeniería.....</b>	<b>68</b>
<b>Anexo 3: Casos de prueba. ....</b>	<b>73</b>

## Introducción

La informática, como ciencia relativamente joven, ha tenido gran desarrollo en el mundo, su uso para lograr la automatización de diversos procesos en las instituciones es uno de los principales aportes que hace esta ciencia. Cuba, enmarcada en un continuo proceso de mejora económica e informatización de la sociedad cubana, ha optado por hacer uso de la informática como herramienta para la mejora de los procesos internos. La Universidad de las Ciencias Informáticas (UCI) juega un papel primordial en esta estrategia seguida por el país, contribuyendo directamente en la informatización de la sociedad cubana y la formación de nuevos profesionales aptos para contribuir con esta. La casa de altos estudios cuenta con doce centros de desarrollo cuya misión es desarrollar proyectos, productos y servicios, garantizando la integración y reutilización de aplicaciones y componentes, de acuerdo con las proyecciones y políticas del país; a la vez que propician la integración de los procesos de formación, producción e investigación, con elevado nivel de compromiso, ética y profesionalidad. (UCI, 2018).

Uno de estos centros de producción es el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) adscrito a la Vicerrectoría de Producción, ofrece servicios de consultoría a organizaciones que ejecutan iniciativas de interoperabilidad e integración de sistemas empleando Arquitecturas Orientadas a Servicios en el contexto del desarrollo de su Arquitectura Empresarial que además desarrolla sistemas empresariales y tiene una línea de investigación de Gestión de Proyectos con un producto asociado (la Suite GESPRO).

Con el objetivo de contribuir a la mejora de los procesos de negocio en la UCI, el centro CDAE desarrollo un Sistema de Reservación en el año 2014 el cual fue adicionado al ya existente Sistema de Gestión de Alimentación que tiene como principal objetivo gestionar con mayor control los servicios alimenticios ofertados en la universidad a más de 8200 comensales entre trabajadores y estudiantes este sistema fue desarrollado en el año 2008.

Cuando se creó el sistema satisfacía las necesidades identificadas por la Dirección de alimento. Dicho sistema lo componen cuatro subsistemas: Reservación, Distribución, Control de acceso y Reportes. El subsistema de Reservación fue desarrollado en el año 2014 a raíz de las nuevas necesidades identificadas por la dirección de alimentos desarrollado sobre la plataforma *Java* siguiendo con la política de *software* libre establecida por la UCI, su principal función es la

reservación por parte de los trabajadores y profesores de los eventos que se realizan en los comedores así como consultar los platos que se ofertaran en el día, lo que contribuye a una planificación más real de estos eventos contribuyendo al ahorro de los recursos alimenticios de la universidad. El subsistema de Distribución tiene como objetivo distribuir a todos los comensales de la institución por uno de sus tres complejos comedores y dentro de uno de estos, el comedor y la puerta por la cual el comensal recibirá el servicio. El subsistema de Reportes se encarga de la emisión de varios reportes a la dirección de alimentación para así mantener el monitoreo y control de la prestación de sus servicios.

El subsistema de Control de Acceso es el encargado de negar o conceder el servicio a los usuarios de este a través de credenciales. Las credenciales han sido utilizadas para la identificación de las personas por más de un siglo, más recientemente volviéndose automatizada debido a los avances en las capacidades de la computación. El subsistema se desarrolló sobre la plataforma *.Net* que por una parte no cumple con la política de *software* libre de la universidad, lo cual implica gastos adicionales por la compra de licencia de *software*, además que el uso de *software* libre da libertad para ejecutar el programa para cualquier fin, estudiar su funcionamiento y adaptarlo, para distribuir copias de la versión que se haya modificado o mejorado; y por otra no permite la interacción de manera óptima con el resto de los subsistemas, estando enmarcado en un sistema donde existe toda una variedad de tecnologías. En el momento que fue creado cumplía con las necesidades identificadas, pero han existido cambios en el sistema de alimentación de la universidad entre los que se encuentra la reservación de los trabajadores de la institución y el cobro de los servicios a estos, estos problemas que se han identificado con el funcionamiento de este subsistema exigen una evolución o remplazo del mismo. Existen oportunidades de mejora con respecto a la sincronización del sistema y su servidor al reestablecerse la conexión perdida por fallas de conexión o pérdida de fluido eléctrico, lo cual provoca la pérdida de información que en muchos casos afecta el funcionamiento de los comedores ya que se le puede dar servicio a comensales doble o dejar de dar servicio a aquellos que no lo han recibido. La confiabilidad de este subsistema también se encuentra comprometida puesto que esta toma la hora de la PC cliente donde se encuentra alojado para determinar el inicio y fin de cada evento del día, el cambio de esta hora en la PC implicaría un cambio en muchos casos no autorizado en el sistema.

### **Problema de la investigación:**

¿Cómo contribuir a la actualización tecnológica del Sistema de Gestión de Servicios de Alimentación desde el subsistema de control de acceso a los comedores?

**Objeto de investigación:**

Sistemas de Control de Acceso.

**Campo de acción:**

Sistemas de control de acceso para comedores.

**Objetivo general:**

Desarrollar un subsistema de control de acceso a los comedores que contribuya a la actualización tecnológica del Sistema de Gestión de Servicios de Alimentación.

**Tareas de la investigación :**

1-Estudio de la documentación de aplicaciones homólogas.

2-Desarrollo de un subsistema de control de acceso.

3-Realización pruebas de validación al software.

**Métodos teóricos:**

- **Analítico – Sintético:** Permitió organizar toda la información obtenida en las bibliografías consultadas referente al tema de control de acceso, para hacer uso de la información adecuada para la realización del presente trabajo.
- **Análisis Histórico – Lógico:** Posibilitó realizar un estudio de la evolución de todo lo referente al control de acceso y del nivel que han alcanzado los sistemas desarrollados en el mundo referente al tema.
- **Modelación:** Permitió hacer una representación, mediante la elaboración de diagramas especificados por la metodología que se utilizará para guiar el proceso de desarrollo del software.

### **Método empírico:**

- **Entrevista:** Permitted interactuar con el personal que tiene conocimiento sobre el tema control de acceso, para aumentar el conocimiento sobre esta medida de seguridad.

### **Idea a defender:**

Con el desarrollo de un nuevo subsistema de control de acceso a los comedores en la tecnología *Java* se lograría la unificación de éste con el resto de subsistemas mejorando su interacción, se alinearía con la política de software libre de la UCI.

### **Aporte práctico de la investigación:**

Se obtiene un subsistema de control de acceso a los comedores desarrollado en un lenguaje libre (*Java*) capaz de interactuar de forma satisfactoria con el resto de subsistemas y que maneja las interrupciones de comunicación entre el cliente y el servidor.

### **Estructura de la tesis:**

Capítulo 1: *Fundamentación teórica*. Se realiza un análisis sobre el estado del arte de las principales implementaciones del Control de Acceso a comedores, de cada una de las tecnologías, herramientas y metodologías a emplear durante la investigación.

Capítulo 2: *Propuesta de solución*. Se aborda al análisis y diseño del sistema cumpliendo con los patrones de diseños y de arquitectura que se seleccionaron para el desarrollo del sistema. Se generarán los artefactos necesarios durante el ciclo de vida del proceso de desarrollo, especificando las características del sistema, desglosadas en escenarios y requisitos de calidad del servicio. Además se diseña el diagrama de clases.

Capítulo 3: *Validación y pruebas*. Se describen las características de la solución propuesta, cumpliendo con los estándares de codificación para el desarrollo del sistema, se muestran los diagramas de componentes y despliegue, así como la descripción de las pruebas realizadas al sistema.



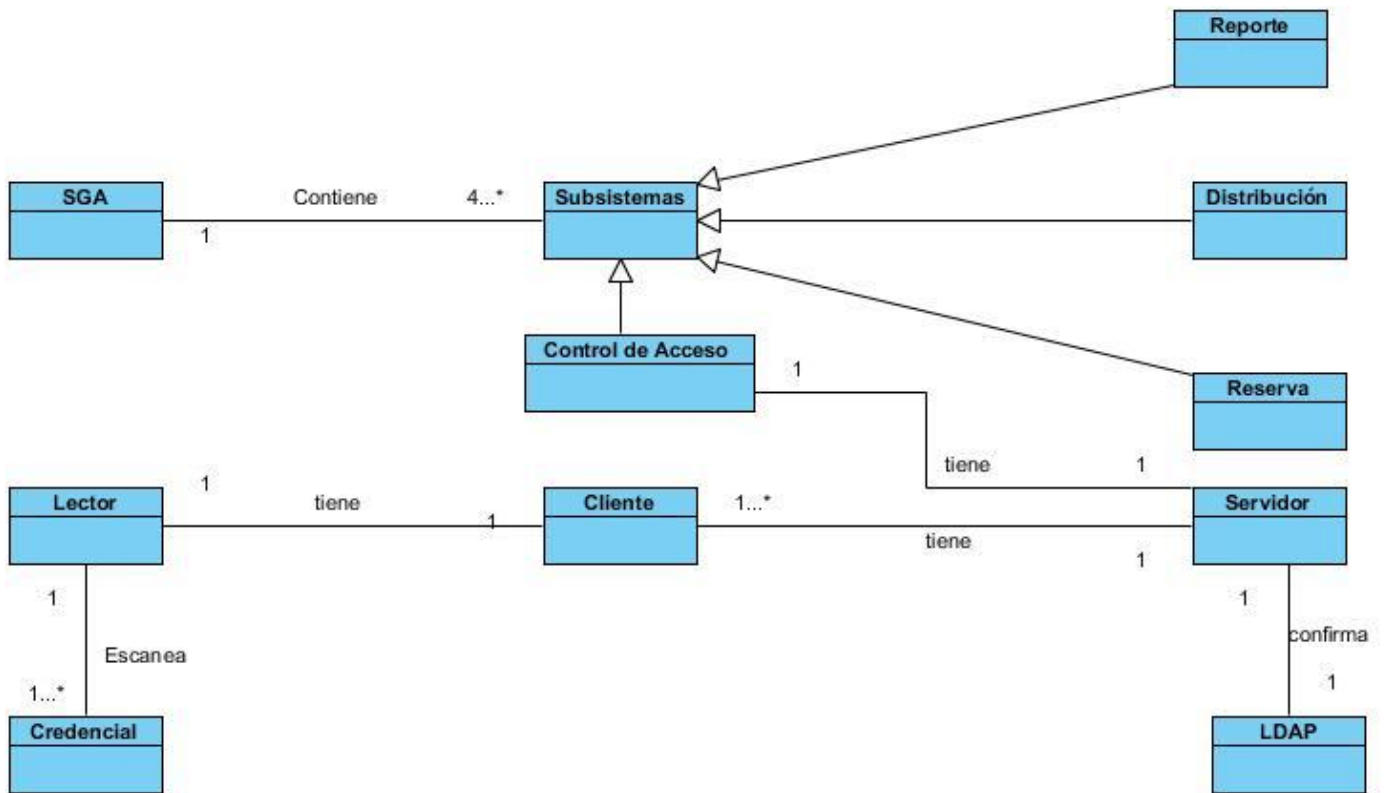
## **CAPÍTULO1: Fundamentos de la investigación, metodologías y tecnologías empleadas.**

### **Introducción**

En el presente capítulo se muestran los resultados de la investigación realizado sobre el estado del arte de los sistemas informáticos de control de acceso. Haciendo referencia a los principales conceptos relacionados con el tema, que son necesarios para un correcto entendimiento del objetivo y el alcance del subsistema a desarrollar. Se hace un estudio de las principales metodologías de desarrollo de software y tecnologías utilizadas para el desarrollo de la solución.

#### **1.1 Sistema de alimentación UCI.**

Actualmente en la UCI se encuentra en explotación un subsistema de control de acceso en los comedores, el mismo está compuesto por un servidor que se comunica con los subsistemas de reservación y el de distribución para conceder acceso a los comensales, una PC cliente en cada puerta de acceso, que además tienen un lector de código de barras que obtiene información de la credencial UCI (Solapín) que es un método de identificación único creado en la universidad. Para mayor entendimiento se presenta el siguiente diagrama:



**Ilustración 1: Funcionamiento del sistema de alimentación.**

SGA: Sistema de gestión de servicios de alimentación. Gestiona todos los servicios ofertados en la universidad dígame desayuno, almuerzo o comida.

Subsistema : Un conjunto de elementos interrelacionados que, en sí mismo, es un sistema, pero a la vez es parte de un sistema superior.

Reporte: Subsistema del Sistema de gestión de servicios alimenticios, encargado de generar reportes a la Dirección de Alimentación.

Distribución: Subsistema del Sistema de gestión de servicios alimenticios, encargado de distribuir a los comensales por los complejos comedores de la universidad.

Reserva: Subsistema del Sistema de gestión de servicios alimenticios, encargado de proporcionar la reserva por parte de los trabajadores de la UCI.



Control de acceso: Subsistema del Sistema de gestión de servicios alimenticios, encargado de negar o conceder el acceso a los comedores.

Credencial: Las credenciales de identificación son tarjeta con fotografía digital y los datos de la persona a identificar. También pueden llevar código de barras o banda magnética que, junto a los respectivos lectores, se utilizan como sistema de acceso y control.

Lector: Un lector de códigos de barras es un dispositivo electrónico que por medio de un láser lee el código de barras y emite el número que muestra el código de barras.

Cliente: PC cliente donde se encuentra alojado el subsistema de control de acceso.

Servidor : Servidor del subsistema de control de acceso.

LDAP: Es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio para verificación de usuarios UCI.

## **1.2 Sistema de control de acceso basado en credenciales.**

El sistema de control de acceso es una red coordinada de tarjetas de identificación, lectores electrónicos, bases de datos especializadas, software y computadoras diseñadas para monitorear y controlar el tráfico a través de puntos de acceso.

Los sistemas de control de acceso basados en tarjetas inteligentes son una herramienta de seguridad poderosa, eficiente para proteger los bienes de una empresa. A cada empleado o contratista se le emite una tarjeta de identidad inteligente que muestra la información de la empresa y diseños impresos, tanto para limitar la posibilidad de falsificación como para identificar que la tarjeta es oficial. Generalmente, la tarjeta muestra una foto de su portador. Cada tarjeta almacena información protegida sobre la persona y sobre los privilegios de esta persona. Cuando la persona se registra inicialmente y acepta la tarjeta, estos privilegios son diseminados (*populated*) a través de todo el sistema de forma veraz y segura (Si tales privilegios cambian, la nueva información puede ser inmediatamente actualizada de manera segura a través de la red).

Cuando la tarjeta es colocada dentro o cerca de un lector electrónico, el acceso se brinda o se niega de forma segura y precisa a todos los espacios adecuados (por ejemplo, un campo, un garaje de estacionamientos, un edificio o una oficina). Cuando un empleado deja la organización, todos los privilegios de acceso físico son removidos de una sola vez. Cualquier tentativa futura por esta persona de reingresar al establecimiento usando una tarjeta expirada o revocada, puede ser negada y este hecho registrado automáticamente (SCALA, 2003).

Un sistema de control de acceso brinda protección tanto para el acceso físico como lógico de forma simultánea. La credencial usada para el acceso físico puede también permitir acceso a la red de computadoras e ir a una infraestructura de clave pública PKI por sus siglas en inglés *Public Key Infrastructure* incluyendo el uso de acceso remoto seguro, correo electrónico seguro, firma digital y red privada virtual segura VPN por sus siglas en inglés *Virtual Privated Network* La meta de protección simultánea puede ser alcanzada por la mezcla o por compartir bases de datos seguras dedicadas a cada tipo de aplicación, permitiendo tanto un control administrativo centralizado como el análisis de cualquier tentativa de acceso no autorizada. Al combinarse la información de monitoreo tanto del sistema físico como lógico, permitirá que las políticas de seguridad puedan ser cumplidas e investigadas a todos los niveles. La información recolectada puede ser invaluable en el análisis de riesgo del conjunto de la empresa (TECNOSeguro, 2014).

La adopción de sistemas de control de acceso basados en tarjetas inteligentes puede también resultar en otras ventajas para la organización, incluyendo:

- Eliminación o reducción de la necesidad de múltiples tarjetas, PINs, o códigos de acceso.
- Apuntalar sistemas preexistentes, de una forma costo eficiente, incluyendo la re-utilización de algunos componentes del sistema de acceso físico, mientras se logra un aumento significativo en seguridad.
- Eliminación de la necesidad de reemplazar las tarjetas cuando los derechos o privilegios cambian.
- Administración centralizada, permitiendo a la organización mantener o aumentar la seguridad mientras se ahorra tiempo, logrando una distribución más completa de la

información, manejando cambios globales para privilegios de acceso a partir de un único punto y reduciendo las complejidades involucradas en la sincronización de sistemas múltiples (ecured, 2018).

### **Visión general de sistemas de control de acceso.**

Para el usuario, un sistema de control de acceso está compuesto de tres elementos:

- Una tarjeta o ficha (una credencial de identidad) que se presenta al lector de la puerta de acceso.
- Un lector que indica si la tarjeta es válida y se autoriza la entrada.
- Una puerta de acceso o portón, que es destrabado cuando se autoriza la entrada.

Detrás de la escena existe una red compleja de datos, computadoras y software que incorporan una funcionalidad robusta de seguridad. Esta sección describe la operación y componentes de un sistema típico de control de acceso físico basado en tarjetas inteligentes. Él brinda un contexto para entender como tecnologías de tarjetas inteligentes de contacto y sin contacto son usadas en una aplicación de control de acceso (SCALA, 2003).

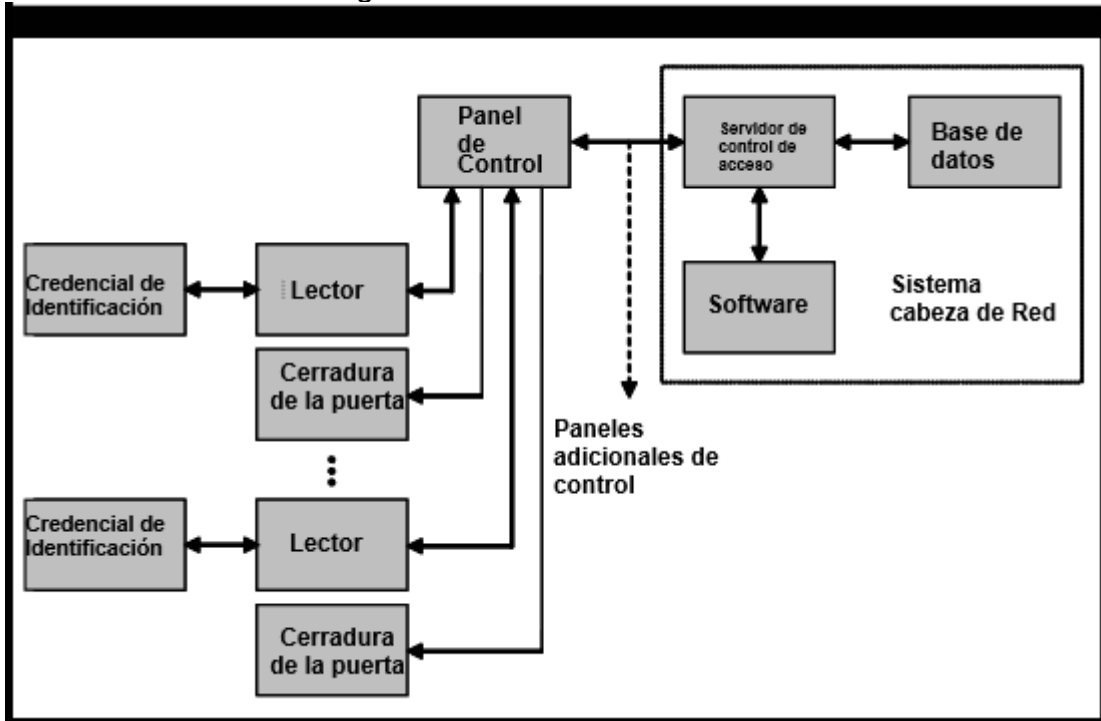
### **1.3 Componentes del sistema de control de acceso.**

Un sistema de control acceso típico está compuesto de los siguientes componentes:

- Una credencial de identificación (tarjeta inteligente).
- Un lector de puerta de acceso (lector de tarjeta inteligente)
- Cerradura de Puerta
- Panel de Control
- Servidor de control de acceso

- Software
- Base de Datos

La Ilustración 2 muestra cómo estos componentes básicos están interconectados. Cada componente será descrito en las siguientes secciones.



**Ilustración 2: Esquema de un Sistema de Control de Acceso tomada de SCALA.**

#### 1.4 Proceso de control de acceso.

Los procesos de control de acceso empiezan cuando un usuario presenta la credencial (típicamente la insignia o identificación de tarjeta inteligente del empleado) al lector, que normalmente está montado próximo a la puerta o portal de entrada. El lector extrae los datos de la tarjeta lo procesa y lo envía al panel de control.

En primera instancia el panel de control valida el lector y luego acepta los datos transmitidos por el lector. Lo que ocurre luego depende sí el sistema es centralizado o distribuido.

En un sistema centralizado, el panel de control transmite los datos al servidor de control de acceso. El servidor de control de acceso compara los datos recibidos de la tarjeta con la información sobre el usuario que está almacenado en la base de datos. El programa de control de acceso determina los privilegios de acceso del usuario y su autorización, la hora, la fecha y la puerta a la que se va a ingresar y cualquier otra información que la compañía pueda requerir para asegurar su seguridad. Cuando se autoriza el acceso, el servidor de control de acceso envía una señal al panel de control para permitir el acceso a la puerta.

En un sistema distribuido, el panel de control permite o niega la entrada. El servidor de control de acceso periódicamente provee datos al panel de control, que habilita al software del panel de control a determinar si el usuario está autorizado o no para tener acceso. El panel de control, entonces, realiza las funciones del servidor de control de acceso descrito arriba y toma la decisión de permitir o negar la entrada. El habilitar el panel de control para realizar la función de decisión tiene la ventaja de requerir menor comunicación entre los paneles de control y el servidor de control de acceso central, mejorando el desempeño y la confiabilidad del sistema como un todo.

Si una función biométrica o un PIN se incorpora al sistema, el lector autentica estos datos. La validez puede ser determinada por el lector o desde dentro de la misma tarjeta inteligente al comparar el dato con un patrón biométrico o un PIN almacenado en la tarjeta. (En algunos casos los datos biométricos pueden ser enviados al panel de control para su procesamiento). Si la información adicional es válida, el lector envía el número de identificación de la credencial al panel de control. Si la información no es válida, entonces el lector de la tarjeta indica que la entrada es negada. (SCALA, 2003).

## **1.5 Descripción de los componentes del proceso de control de acceso.**

### **La credencial de identificación.**

Una amplia gama de tecnología de identificación está actualmente siendo usado para control de acceso: cintas magnéticas, cintas Wiegand, Bariun Ferrite, tecnología de proximidad de 125KHz las tarjetas inteligentes de contacto y sin contacto. Esas tecnologías pueden ser empaquetadas en diferentes formatos- desde un llavero o una insignia del empleado o incluso formas más

exóticas, como un reloj de pulso o un anillo. Sin embargo, todas las credenciales operan básicamente de la misma forma: ellos almacenan datos que autentican la credencial y/ o el usuario.

Algunas tecnologías de credencial son solamente para lectura. La información está registrada permanentemente en la credencial, y cuando la credencial se presenta al lector la información es enviada al sistema. Ese tipo de credencial solo valida que la información misma es auténtica. Pero no confirma que la persona que está presentando la credencial es la persona autorizada a poseerla, o que la credencial misma es legítima.

La tecnología de tarjeta inteligente de contacto definido, por ISO/IEC 7816 y la tecnología de tarjeta inteligente sin contacto definido, por ISO/IEC 14443 e ISO/IEC 15693, tienen capacidad tanto para leer como escribir y almacenar datos. Las credenciales que usan estas tecnologías son dispositivos inteligentes. Ellos pueden almacenar privilegios, autorizaciones y registros de asistencia. Ellos pueden almacenar los PINs y los patrones biométricos, ofreciendo una capacidad de autenticación de dos o tres factores simultáneamente. La credencial ya no es solo un portador de un número único; pero pasa a ser también, cargador seguro y portátil de datos. (Identificaciones, 2013).

### **El lector de código de barra**

El lector de la puerta de acceso puede tener una o más interfaces, acomodando algún tipo de combinación tanto de las tarjetas inteligentes de contacto como sin contacto e incluyendo un teclado de PIN y un lector biométrico. Como el lector responde va a depender del tipo de credencial presentada y de la política de seguridad de la organización. (Informática, 2018)

Cuando el lector es usado con una tarjeta inteligente sin contacto, él actúa como un pequeño transmisor y receptor de radio de baja frecuencia, constantemente transmitiendo o un campo de radio frecuencia (RF) o un campo electromagnético que es llamado campo de recepción. Cuando la tarjeta sin contacto está dentro del radio de acción del campo de recepción, la antena interna de la tarjeta convierte la energía del campo en electricidad la cual le provee energía al chip que está en la tarjeta. El chip entonces usa la antena para transmitir los datos al lector. (panduit, 2014)

Cuando el lector es usado con una tarjeta inteligente de contacto, el lector incluye una ranura que contiene un contacto de tarjeta inteligente. La tarjeta y el conector (conector) dentro del lector deben hacer contacto (*contact*) físico.

Los lectores que incluyen un teclado de PIN o un lector biométrico (típicamente una huella digital o un lector geométrico de mano), generalmente se respaldan en una autenticación de dos o tres factores, según sea requerido. Por ejemplo, un establecimiento puede requerir solamente la presentación de una tarjeta sin contacto cuando el riesgo de seguridad es bajo, pero puede pasar a requerir datos biométricos también cuando aumenta el nivel de amenaza. Cuando el riesgo de seguridad es alto, puede ser necesario presentar una tarjeta inteligente de contacto, usar un lector biométrico y el teclado de PIN. Estos lectores de factores múltiples pueden ser usados cuando se desea cambiar los insumos requeridos según la hora del día, el día de la semana o la localización. Los requerimientos para los factores adicionales de autenticación han de ser establecidos por la política de seguridad de la organización.

Cuando el lector ha recibido todos los datos requeridos, él típicamente procesa la información en una de dos formas. O la información es enviada inmediatamente al panel de control o el lector analiza los datos antes de enviarlos al panel de control. Ambos métodos son ampliamente utilizados, cada uno tiene sus ventajas y desventajas.

Los lectores más sencillos envían los datos directamente al panel de control. Esos lectores no hacen nada para evaluar los datos o determinar la legitimidad de la credencial. Esos lectores son típicamente lectores de un solo factor y son genéricos, así que pueden ser almacenados en inventarios y fácilmente adicionados o intercambiados en un sistema de control de acceso.

Los lectores que analizan los datos deben estar integrados en el sistema de control de acceso. O sea, ellos deben interpretar y manipular la información enviada por la tarjeta y entonces transmitir los datos en un formato que pueda ser usado por el panel de control. Tal sistema puede ofrecer un nivel incrementado de seguridad. El lector puede determinar la legitimidad de la tarjeta (y la tarjeta puede determinar la legitimidad del lector) comparar los datos biométricos o la entrada de un PIN y manipular los datos de la credencial ya que lo que el lector envía al panel de control no es lo mismo que fue leído de la tarjeta. El proceso de autenticación de la tarjeta al lector y del

lector a la tarjeta es llamado autenticación mutua. Autenticación mutua es una de las ventajas del sistema basado en tarjetas inteligentes. (SCALA, 2003).

## **Panel de Control**

El panel de control (frecuentemente conocido como el controlador o simplemente el panel) es el punto central de comunicaciones para el sistema de control de acceso. El panel de control típicamente supe energía y establece interfaces con múltiples lectores en diferentes puntos de acceso. El panel o controlador conecta con la cerradura electromecánica de la puerta de acceso necesario para físicamente desatranca la puerta o el mecanismo para un portón de entrada (tal como un sistema rotatorio o un portón de estacionamiento o un elevador). El panel puede estar conectado a diferentes alarmas (por ejemplo, sirenas, digitalizadores automáticos, luces). Y finalmente, el panel de control generalmente está conectado a un servidor de control de acceso.

Dependiendo del diseño del sistema, el panel de control puede procesar datos del lector de tarjetas y del servidor de control de acceso y tomar la decisión última sobre autorización o él puede pasar los datos al servidor de control de acceso para que él tome esa decisión. Típicamente, el panel de control toma la decisión de desatranca la puerta, pasa los datos de esa transacción al computador base y envía una señal de desbloquear hacia el lector. Es importante que sea el panel de control (y no el lector) el que genere la señal de desatranca, ya que el panel de control está localizado dentro del establecimiento en un cuarto seguro, mientras el lector de la tarjeta está localizado en un área insegura o abierta. Finalmente, el panel de control almacena información sobre los formatos de datos. Esa información identifica que porción del flujo de datos recibidos de una tarjeta es usada para tomar decisiones de control de acceso. Tarjetas y lectores con diferentes tecnologías pueden intercambiar datos en diferentes formatos. Sin embargo, el panel de control necesita saber como interpretar y procesar estos datos. Por ejemplo, si un lector envía 35 bits de *data* y el panel de control está diseñado para leer solamente 26 bits, el panel debe rechazar los datos o trunca 9 bits. El formato de los datos controla como el panel interpreta los datos recibidos. (SCALA, 2003).

## **El Servidor de Control de Acceso**



El sistema de cabeza de red (también conocido como sistema de “*back end*”, sistema huésped “*Host system*”) incluye el servidor de control de acceso, el *software* y una base de datos. La base de datos contiene información actualizada sobre los derechos de acceso de los usuarios. En un sistema centralizado, el servidor de control de acceso recibe los datos de la tarjeta del panel de control. El *software* correlaciona los datos de la tarjeta con los datos en la base de datos, determina los privilegios de acceso de la persona, e indica si la persona puede o no ser admitida. Por ejemplo, si una persona está autorizada a ingresar a un edificio solamente entre las 8:00 a.m. y 5:00 p.m. y son las 7:45 a.m., está persona no puede ser admitida. Sin embargo, si son las 8:01 a.m., entonces el computador debe responder al panel de control, indicando que la puerta puede ser abierta.

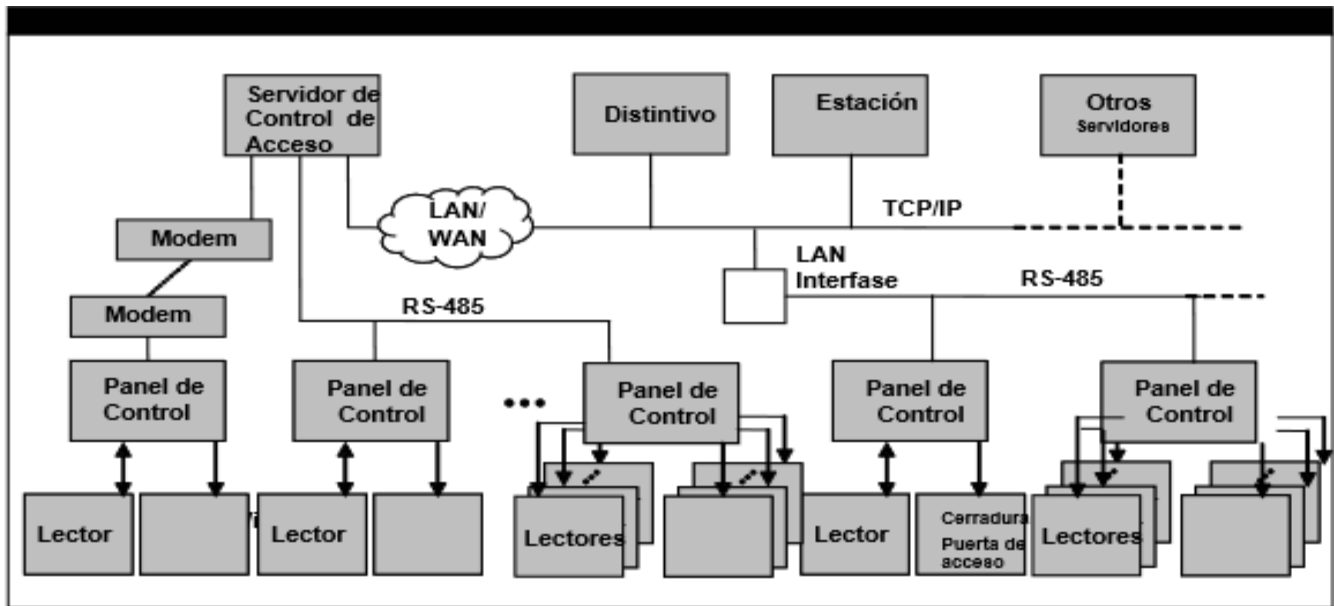
La mayoría de los sistemas son descentralizados. En un sistema descentralizado, el servidor de control de acceso periódicamente envía información de control de acceso actualizada a los paneles de control y les permite operar independientemente, tomando la decisión de autorización para las credenciales presentadas basadas en los datos almacenados en el panel.

Las características operacionales en los sistemas centralizados o descentralizados, son determinadas por los requerimientos específicos de implementación de control de acceso de la organización. (SCALA, 2003).

## **1.6 Implicaciones de las Tendencias Recientes en la Arquitectura de Sistemas.**

Los sistemas de control de acceso tradicionalmente han sido controlados por el departamento de seguridad corporativa. Sin embargo, con el advenimiento de sistemas corporativos centralizados, basados en tecnología de internet y de TCP/IP, los sistemas de control de acceso han evolucionado a sistemas de red que combinan muchas funciones e involucran múltiples departamentos. Sistemas modernos pueden incluir no solo funciones de control de acceso, sino también funciones corporativas tales como: manejo de credenciales y bases de datos personales. Tampoco los sistemas de control de acceso en redes han alcanzado sus límites funcionales: es fácil concebir un lector de tarjeta que actué como un reloj que marca el tiempo de entrada y salida de los empleados, extendiendo así el sistema al departamento de recursos humanos y de pagos

(Ilustración 3) o una tarjeta de identificación que incluye una aplicación de pago para el sistema local de transporte.



**Ilustración 3: Ejemplo de un Sistema de Control de Acceso Físico.**

### 1.7 Sistemas de control de acceso estudiados a nivel internacional.

#### **Sistema de control de acceso *Arquero***

El sistema de control de acceso *Arquero* puede controlar el acceso a las diferentes dependencias de una empresa. No solo permite regular el paso de los empleados, sino que es capaz de realizar tareas de vigilancia. Además, la capacidad de la herramienta para generar informes de eventos de identificación, administración, topología, recintos etc. hace que el mismo sea una herramienta sumamente útil a la hora de extraer históricos sobre cualquier evento ocurrido en la empresa.

#### **Sistema de control de acceso *Easy Way***

El sistema de control de accesos *Easy Way* es un método destinado a controlar el ingreso y egreso de personas a todas las áreas de la empresa. El software de control de acceso permite configurar el hardware desde la PC, controlar desde la inclusión de planos del edificio hasta generar informes y elaborar estadísticas. Este software de control de personal tiene los controles

de acceso totalmente integrados y en forma modular, es adaptable a sofisticados requerimientos particulares que puedan llegar a solicitarse, estableciéndose así, una relación personalizada con el cliente y el *software* de control de accesos. Esta unidad de control de acceso trabaja en forma autónoma tanto para la apertura de puertas, como para accionar molinetes, barreras, alarmas, portones y sistemas de cacheo. Estos controladores se adaptan a cualquier sistema de lectura: banda magnética, códigos de barras con filtro infrarrojo, proximidad, biometría como lectores de huellas digitales y lectores de proximidad HID. (Yasmani Freixas Ramírez, 2015).

### **Sistema de control de acceso *BioStar VideoPhone***

BioStar es el nombre del sistema de control de acceso con conectividad de IP y seguridad biométrica del reconocido fabricante *Suprema Inc.* *BioStar VideoPhone* es una aplicación para PC que permite al operador utilizar la PC y un dispositivo ligado como un sistema de interfono. La versión *BioStar VideoPhone* es un software para intercomunicación por video basado en IP que reemplaza un videoteléfono análogo. Integrado con lectores *Suprema* de modelos seleccionados, la aplicación permite al operador visualizar quién está en la puerta y permitir el acceso, si este se ha aprobado. Es una aplicación única y muy útil empleada para crear un sistema de intercomunicación por video sin costos o equipo de videoteléfono adicional. Para permitir que esta característica se encuentre ampliamente disponible, la instalación/configuración y uso del sistema fueron diseñados para la conveniencia y simplicidad del usuario. (Yasmani Freixas Ramírez, 2015)

### **Sistema de control de acceso *Digital Persona Pro Software***

*Software* de Administración de claves de acceso mediante el uso de la huella dactilar. El *software Digital Persona PRO* es un sistema sólido de administración de usuarios y *passwords* mediante el uso de la huella dactilar, logrando autenticación multifactor que brinda la mayor seguridad posible a sus redes, servidores y aplicaciones administrativas. Puede usarse en ambientes de red, integrándose al "*Active Directory*" de Windows. *Digital Persona PRO* brinda a las organizaciones una forma eficiente en costo de aumentar su seguridad en tecnologías de información y simplificar normativas de control. (Yasmani Freixas Ramírez, 2015)

### **Sistema de control de acceso *ExClouds***

*ExClouds* es una aplicación basada en la web del Sistema *ExpansE10*. Permite una configuración más rápida y fácil; no requiere instalación de *software* ni configuración de la

computadora. Controla todas las opciones de configuración del sistema, así como los servicios adicionales, inclusive vigilancia en tiempo real desde cualquier panel y computadora en la red. La aplicación de servidor de *ExClouds* complementa el sistema con capacidades avanzadas de generación de informes. La interfaz web de *ExpansE* es fácil de usar y facilita la gestión de control de acceso, video y monitoreo de alarmas desde cualquier panel o navegador de Internet, y no se requiere operadores expertos. Además, la arquitectura completa de cliente-server permite el uso de múltiples-clientes a niveles diferentes de usuario. (Yasmani Freixas Ramírez, 2015)

## **1.8 Sistemas de control de acceso estudiados desarrollados a nivel nacional.**

### **Sistema de Identificación *ExpansE*.**

ExpansE es un innovador sistema distribuido de control de acceso que incorpora una aplicación de control y administración basada en web, utilizando lo último en tecnología para control de accesos que le permiten adecuarse a los requerimientos de un mercado cambiante.

Este sistema brinda un servicio de certificación de identidad a otros sistemas informáticos, como los destinados al control del acceso. Tiene almacenados los datos de todo el personal que labora y estudia en la UCI: estudiantes y todo tipo de trabajadores. Lo más importante es que le asigna a cada persona un código único, para su identificación. Este sistema está estructurado por los siguientes módulos; administración, configuración, identificación, detección de rostros y seguridad. Además, utiliza framework como Spring Framework y .Net, posee una arquitectura por capas, dichas características hacen que dicha aplicación sea reutilizable, lo que brinda la facilidad de utilizar módulos tales como seguridad y configuración en la aplicación a desarrollar. (Yasmani Freixas Ramírez, 2015).

### **Sistema de Control de Acceso a Comedores.**

Mediante este sistema se controla en los comedores de las diferentes edificaciones donde se brinda el servicio de alimentación; el acceso de los estudiantes, profesores y trabajadores durante las tres sesiones de servicio: desayuno, almuerzo y comida. El mismo se divide en dos partes: el control de acceso y la gestión de comensales. El acceso se controla registrando el código de barras, que se encuentra en la identificación de cada persona, en cada una de las puertas de los comedores. La gestión de comensales permite a los directivos la asignación de los

comedores y puertas a los mismos, además de ofrecer reportes como la cantidad de comensales desglosado por puerta o tipo. (Yasmani Freixas Ramírez, 2015).

### **Sistema de control de acceso a los laboratorios de producción (UCILAB).**

Este sistema lleva el control de los proyectos que radican en los laboratorios destinados a los procesos productivos y por tanto de las personas que pueden tener acceso a dichos laboratorios. En este sistema se chequea qué personas tienen acceso o no a los laboratorios, verificando que estén en la base de datos correspondiente, mediante el número de la identificación. Existen varias implementaciones de este sistema en la UCI, cada una de ellas específica para el área productiva donde se encuentra, lo que hace que no exista una base de datos centralizada con todos los datos referentes a todos los laboratorios de producción. Sin embargo, la aplicación a desarrollar debe ser capaz de gestionar toda la información que manejan dichas soluciones de forma centralizada. (Yasmani Freixas Ramírez, 2015).

### **Sistema de gestión de laboratorios (COSMO).**

Sistema capaz de controlar los aspectos relacionados con la interacción de los usuarios y los laboratorios o agrupaciones de computadoras, tales como la entrada y salida de los usuarios, las horas de inicio y fin de sesión de las computadoras, lista de procesos activos, entre otros. Además, COSMO permite generar estadísticas de apoyo a la toma de decisiones. Fue desarrollado en el lenguaje C# y está diseñado preferentemente para sistemas operativos propietarios. (Yasmani Freixas Ramírez, 2015).

### **Sistema de control de acceso para el centro CISED en la UCI.**

El sistema se encuentra actualmente en funcionamiento, en el centro de desarrollo de software CISED, con el objetivo de controlar el personal que accede a los laboratorios asignados a la producción. La funcionalidad principal del controlador de acceso es impedir que el usuario pueda acceder a los servicios de red si este no se ha registrado por el controlador de entrada, aplicación web que recibe el número de solapín. El sistema no puede controlar las aplicaciones iniciadas por el usuario en las estaciones de trabajo. *Jkeeper* Sistema de control de acceso automatizado para los laboratorios de la Universidad de las Ciencias Informáticas, el sistema permite el control de acceso de los usuarios a los laboratorios, el control de las estaciones de trabajo en cuanto a sesiones y aplicaciones iniciadas, mantiene un control estricto sobre los accesos que acontecen en la universidad y el control de las estaciones de trabajo, elevando el control sobre la adecuada

explotación de las estaciones de trabajo, así como la generación de reportes a partir de la información obtenida de las funcionalidades descritas, para la toma de decisiones administrativas. (Yasmani Freixas Ramírez, 2015).

### **Sistema de control de acceso e interbloqueo para el Centro de Inmunología Molecular.**

Sistema desarrollado en la CUJAE compuesto por dos elementos fundamentales: el controlador de puerta y el controlador de interbloqueo. Ambos son tarjetas electrónicas con finalidades diferentes dentro del sistema. La función del controlador de puerta es, como su nombre lo indica, el control de las puertas. Un controlador de este tipo puede controlar un máximo de dos puertas. De esta forma se encarga de recibir la solicitud de acceso que viene del elemento de identificación, determinar si el código arribado tiene acceso por la puerta especificada según previa configuración, de ser positiva o válida, entonces desbloquear la puerta en cuestión. Otro elemento asociado al sistema es un dispositivo para detectar el caso de que un usuario desee salir de un local a través de alguna puerta controlada, es decir, por la parte contraria en donde se encuentra situado el lector. Estos casos se denominan *REX (Request for EXit)* en la mayoría de los sistemas de control de acceso, y para ello se incluye en el sistema un elemento *hardware* que notifique, a través de una señal, esta acción al controlador. Este elemento puede ser un sensor de presencia o un simple botón. El controlador de interbloqueo se encarga de agregar la capacidad de interbloqueo al sistema. Este mecanismo consiste en el bloqueo de una o más puertas tras la detección de la apertura de una puerta determinada. Esto se realiza, por supuesto, con una previa configuración del controlador, donde se establece esta relación o interacción entre las puertas. Este sistema puede tener muchos usos o aplicaciones y, en el caso del CIM, se utiliza para disminuir los riesgos de contaminación cruzada, concepto asociado a las GMP en el que una persona saliendo del área limpia pueda inducir contaminación a aquella que está entrando. (Yasmani Freixas Ramírez, 2015).

### **1.9 Tecnologías, metodología, lenguaje y herramientas.**

Luego de analizar la panorámica referente al estado del arte de las soluciones de Control de Acceso existentes, es importante abordar sobre las principales tecnologías, lenguajes,

herramientas que se utilizarán en la realización de este sistema y la metodología que guiará su proceso de desarrollo. Como el sistema que se implementará será un subsistema del sistema de gestión de servicios alimenticios, se deberán utilizar las mismas tecnologías, lenguajes y herramientas que fueron utilizadas en la realización de este sistema, aunque podrían incluirse nuevas según la necesidad que exista para poder darle solución al problema existente.

## **LDAP.**

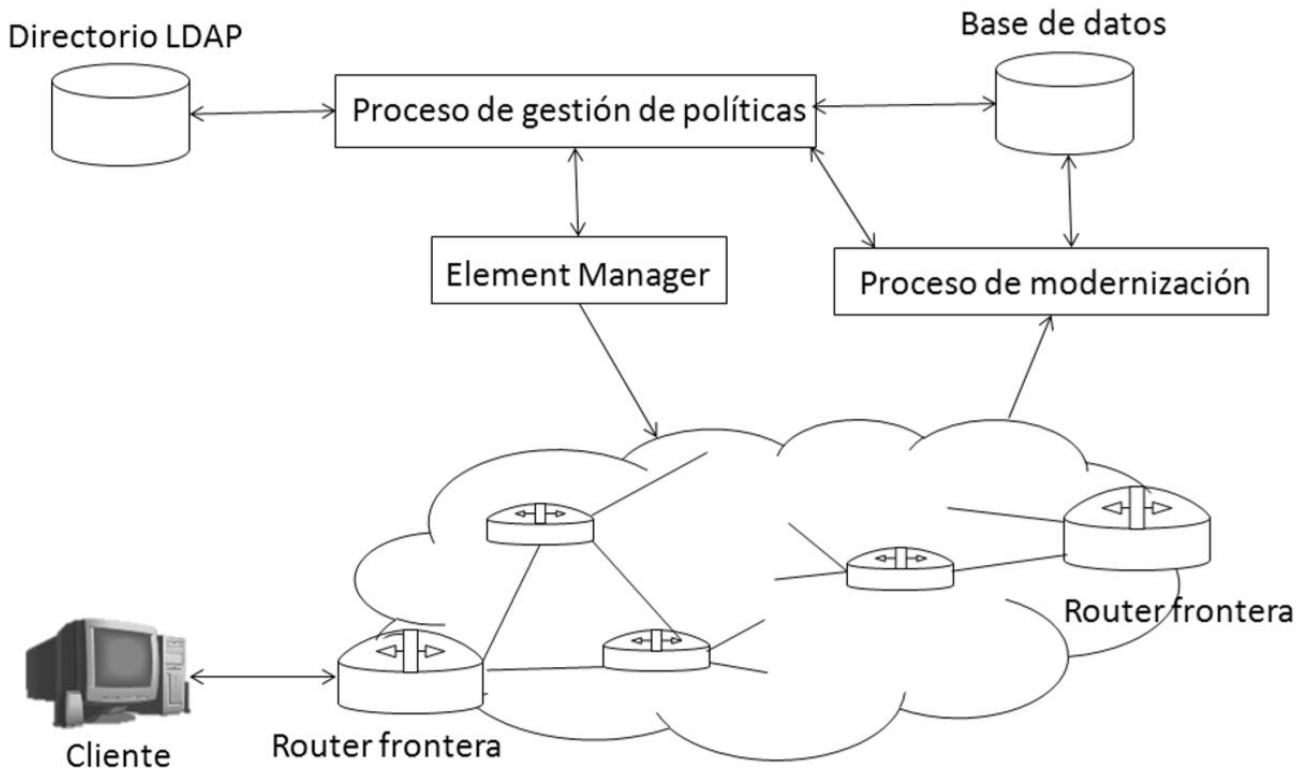
*Lightweight Directory Access Protocol (LDAP*, por sus siglas en inglés), en español Protocolo Ligerero de Acceso a Directorios es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio. LDAP funciona sobre TCP/IP u otros servicios de transferencia orientados a conexión y está basado en el estándar X500. (Timothy A. Howes, 2003).

## **Directorio LDAP.**

Un directorio LDAP es una base de datos optimizada para la lectura y búsqueda de información que es almacenada de manera jerárquica. Los directorios soportan opciones avanzadas de filtrado. Generalmente no soportan transacciones complejas que sí ofrecen los sistemas de gestión de bases de datos diseñadas para procesar un gran volumen de actualizaciones. Los cambios en la información almacenada en un directorio suelen ser del tipo "o todo o nada", es decir, cambios de las ramas del árbol de directorio de información (DIT por sus siglas en inglés) completamente, pero, aunque no estén optimizados para ello, los directorios pueden permitir cambios muy específicos. Los directorios están preparados para dar una respuesta rápida a un gran volumen de búsquedas. Disponen de mecanismos de replicación de la información en varios servidores para incrementar la disponibilidad y fiabilidad del servicio mientras se reduce el tiempo de respuesta.

Un directorio es una base de datos, pero que en general contiene información más descriptiva y basada en atributos de usuarios y recursos de red. Permite hacer una preselección de las políticas mediante la selección de las ramas en función de la información de la red antes de hacer una búsqueda. Ofrece un control dinámico y coordinado de los elementos de la red ya que las decisiones se toman de manera automática basándose en reglas, peticiones de usuarios o de servicios. Esta gestión dinámica de los elementos de la red representa un cambio cualitativo,

desde el punto de vista empresarial, ya que permite la gestión de los recursos de la red y de los servicios de manera más eficiente. (Timothy A. Howes, 2003).



**Ilustración 4:Funcionamiento de LDAP.**

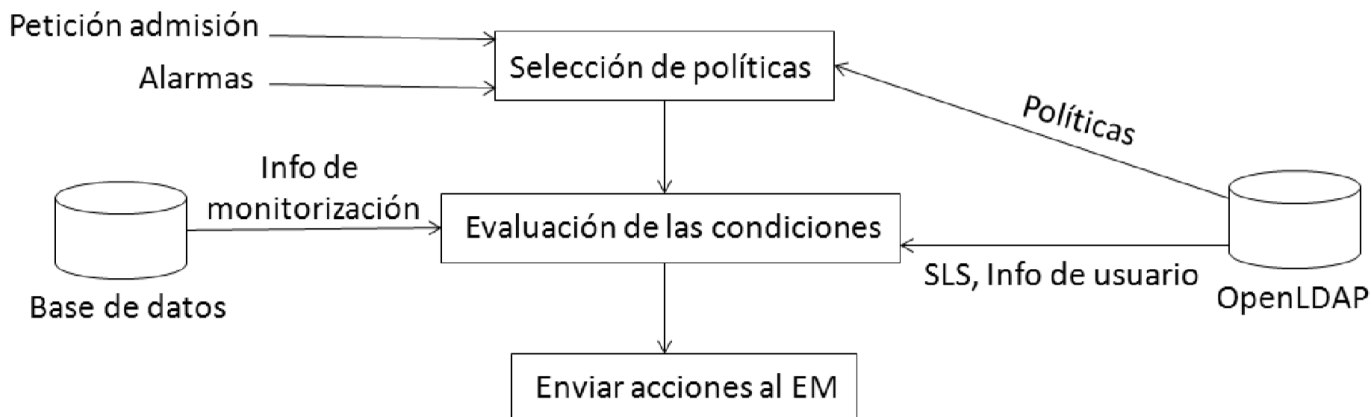
### **OpenLDAP.**

Es una aplicación de servidor *LDAP*, que proporciona la funcionalidad de Servicios de Directorio, tales servicios las incluyen para gestionar identidades y las relaciones entre los ordenadores, usuarios y grupos de ordenadores o usuarios que participan en la red, y proporcionan una forma consistente de describir, localizar y gestionar esos recursos.

El servidor *OpenLDAP* posibilita: Implementación del protocolo *LDAPv3* que está definido para que permita el uso de *IPv4* e *IPv6*. Autenticación y nivel de seguridad que permite servicios de autenticación mediante capa de seguridad y autenticación simple (*SASL*, por sus siglas en inglés). El acceso al directorio ha de ser restringido ya que contiene información de usuarios (nombres de usuarios y contraseñas), información de las políticas y parámetros. Por esta razón sólo el administrador del directorio tiene acceso a la información. Control de acceso: *OpenLDAP*



permite definir una serie de filtros para el control de acceso a diferentes *DITs*, entradas, o atributos de estas entradas. Es totalmente gratuito bajo licencia *Open Source*. El servidor *OpenLDAP* hace una implementación *Open Source* del protocolo *LDAP* y requiere un motor para almacenar y hacer búsquedas de la información.



**Ilustración 5: Funcionamiento de OpenLDAP.**

### Máquina virtual de *Java*

La Máquina Virtual de *Java* (JVM, por sus siglas en inglés) es el entorno en el que se ejecutan los programas *Java*, su misión principal es la de garantizar la portabilidad de las aplicaciones *Java*. Define esencialmente un ordenador abstracto y especifica las instrucciones que este ordenador puede ejecutar. Las tareas principales de la JVM son las siguientes:

- Reservar espacio en memoria para los objetos creados.
- Liberar la memoria no usada.
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos.
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones *Java*.

(SANABRIA, 2002).

### Bootstrap 3.1.1

*Bootstrap* es el *framework* de *Twitter* que permite crear interfaces web con *CSS* y *Javascript* que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una *Tablet* sin que el

usuario tenga que hacer nada, esto se denomina diseño adaptativo o *Responsive Design*. Aun ofreciendo todas las posibilidades que ofrece *Bootstrap* a la hora de crear interfaces web, los diseños creados con *Bootstrap* son simples, limpios e intuitivos, esto le da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El *framework* trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración *jQuery* para ofrecer ventanas y *tooltips* dinámicos.

*Bootstrap* tiene un soporte relativamente incompleto para *HTML5* y *CSS3*, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en *CSS3* para las esquinas redondeadas, gradientes y sombras son usadas por *Bootstrap* a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso. (GetBootstrap.com, s.f.)

## **1.10 Metodologías de desarrollo de software**

“Una metodología de ingeniería de *software* es un proceso para la producción organizada del *software*, empleando para ello una colección de técnicas predefinidas y convencionales en las notaciones. Una metodología se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada uno de ellos. Los pasos de la producción del *software* se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo”. (Chandra, 2015).

Una metodología no es más que un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la realización de un software. Estas tienen como objetivo primordial lograr que los productos finales sean eficientes y que cumplan con los requerimientos planteados por el usuario. Como los requisitos son tan variables y desiguales, han surgido varias metodologías de desarrollo, las cuales han sido clasificadas en dos grandes grupos de acuerdo con sus características y los objetivos que persiguen: robustas y ágiles.

### **Proceso Unificado de Desarrollo (RUP):**

Dentro de las metodologías pesadas se encuentra el Proceso Unificado de Desarrollo (*RUP*, por sus siglas en inglés), que por ser la más completa constituye un ejemplo académico cuando se quieren estudiar estas metodologías. *RUP* cuenta con cuatro fases de trabajo (Inicio, Elaboración, Construcción y Despliegue) divididas en flujos de trabajo, esta estructura hace que el desarrollo con *RUP* sea dirigido por casos de uso, centrado en la arquitectura e iterativo - incremental. Estas metodologías demandan de un numeroso equipo de proyecto y generan una gran cantidad de documentación, que en ocasiones resulta inconveniente para proyectos sencillos y con poco tiempo para el desarrollo, como es el caso que se plantea. Por estas razones se decide desechar este tipo de metodología y analizar las ágiles.

Las metodologías ágiles permiten dar una rápida respuesta a cambios en los requisitos a lo largo del proyecto e ir entregando de forma continua y en plazos cortos software funcional. El trabajo conjunto entre el cliente y el equipo de desarrollo minimiza los costos frente a cambios y eliminar el trabajo innecesario. Estas metodologías prestan gran atención a la excelencia técnica y al buen diseño. Evita malentendidos de requerimientos entre el cliente y el equipo. Además, cada componente del producto final será probado para ver si satisface los requerimientos. (Chandra, 2015).

### **AUP**

El Proceso Unificado Ágil de Scott Ambler, *Agile Unified Process (AUP)* en inglés, es una versión simplificada del Proceso Unificado de Rational (*RUP*). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en *RUP*. El *AUP* aplica técnicas ágiles incluyendo: Desarrollo Dirigido por Pruebas (*Test Driven Development - TDD* en inglés), Modelado ágil, Gestión de Cambios ágil, Refactorización de Base de Datos para mejorar la productividad. Al igual que en *RUP*, en *AUP* se establecen cuatro fases que transcurren de manera consecutiva. (Chandra, 2015).

### **Variación de AUP para la UCI.**

Al no existir una metodología de *software* universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la Metodología *AUP*, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Una metodología de desarrollo de *software* tiene entre sus objetivos aumentar la calidad del *software* que se produce,

de ahí la importancia de aplicar buenas prácticas, para ello se apoyara en el Modelo CMMI-DEV v1.3. El cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad. (UCI, 2018).

### **Metodología a utilizar.**

Para elegir la metodología a utilizar en el proyecto de desarrollo de *software*, se tuvo en cuenta las principales características del mismo:

- Complejidad: Alta.
- Tiempo de desarrollo: 5 meses
- Tamaño del equipo: 1

Una vez que estén bien definidos estos factores, se realiza un análisis preliminar para identificar cuál se ajusta más al proyecto. Se decide utilizar la metodología AUP-UCI que es una adaptación de la metodología AUP en la UCI.

### **Lenguaje de programación.**

Un lenguaje de programación puede definirse como una construcción mental del ser humano para expresar programas. Está compuesto por un grupo de reglas gramaticales, símbolos utilizables, términos con sentido único y una regla principal que resume las demás. Para que esta construcción mental sea operable en un computador debe existir otro programa que controle la validez o no de lo escrito (Almagro, 2011).

### **Java.**

*Java* es un lenguaje orientado a objetos y de alto nivel, fue creado por *Sun Microsystems*. Recoge los elementos típicos en el resto de los lenguajes de programación, agregando variadas ventajas como son la sencillez a la hora de desarrollar en el mismo y la portabilidad de las aplicaciones creadas. *Java* ha sido pensado para soportar aplicaciones que serán ejecutadas en diversos entornos. Esto es posible ya que el mismo genera *bytecode*, usándolos como lenguaje intermedio y siendo indiferente de la arquitectura, el código pasa al intérprete de *Java* el cual

realiza las acciones para que la aplicación sea compatible con las diversas plataformas *hardware* y *software*.

Se utiliza este lenguaje para el desarrollo del subsistema en cuestión debido a que el sistema del que es parte está desarrollado en esta misma tecnología. (SANABRIA, 2002).

## **1.11 Herramientas de desarrollo.**

### **Entornos de desarrollo.**

Un Entorno de Desarrollo Integrado (*IDE*, por sus siglas en inglés), es un programa donde es posible escribir código fuente, compilarlo y ejecutarlo. Permiten desarrollar las aplicaciones de forma mucho más rápida, incorporando en muchos casos librerías con componentes previamente implementados. (Gonçalves, 2008).

### **NetBeans IDE 8.0.**

El *IDE NetBeans* es un entorno de desarrollo integrado disponible para *Windows*, *Mac*, *Linux* y *Solaris*. El proyecto *NetBeans* consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente aplicaciones web, empresariales, de escritorio y aplicaciones móviles utilizando la plataforma *Java*, así como *JavaFX*, *PHP*, *JavaScript* y *Ajax*, *Ruby* y *Ruby on Rails*, *Groovy* y *Grails*, y *C/C++*. *NetBeans IDE 8.0* ofrece un rendimiento significativamente mejorado y la experiencia de codificación, con nuevas capacidades de análisis de código estático en el editor de *Java* y de exploración del proyecto. Esta versión incluye características notables, como es la integración con el generador de escena para crear visuales de *JavaFX*, soportando los múltiples *frameworks* de *PHP* y muchas otras mejoras en *Java EE*, *Maven*, *C/C++* y la plataforma *NetBeans*. (Gonçalves, 2008).

### **IDE a utilizar.**

Se escogió en *NetBeans IDE 8.0* para poder generar los pedidos de certificado utilizando el lenguaje de programación *Java*.

## **1.12 Herramienta de modelado.**

Se decide utilizar *Visual Paradigm* para *UML* 8.0 ya que es una herramienta que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. Además, es una herramienta que presenta versiones con licencias libres.

### **Conclusión parcial.**

- El análisis de las diferentes metodologías, tecnologías y lenguajes según las necesidades de la solución, determinó la utilización de la metodología AUP-UCI, UML como lenguaje de modelado y la herramienta *Visual Paradigm* para el modelado. El lenguaje de programación *Java* usando el *IDE NetBeans*.
- El estudio de los conceptos relacionados con el proceso de Control de Acceso basado en credenciales y el análisis de los sistemas similares existentes, demostró los principales componentes del proceso de control de acceso.

## **CAPÍTULO 2: Descripción de la solución propuesta.**

### **Introducción.**

En este capítulo se muestra la propuesta de solución a la problemática planteada. Se representa el Modelo de dominio donde se capturan los objetos importantes del entorno donde será empleado el sistema. Se identifican los requerimientos funcionales y los no funcionales, se elaboran las historias de usuarios.

### **2.1 Modelo de dominio.**

El modelo del dominio muestra clases conceptuales significativas en un dominio de problema. Es una representación de las clases conceptuales del mundo real, no de componentes de software.

#### **Clases fundamentales del entorno.**

**Administrador:** persona encargada de junto al departamento de economía planificar y registrar en el sistema las raciones diarias de comida a ofertar.

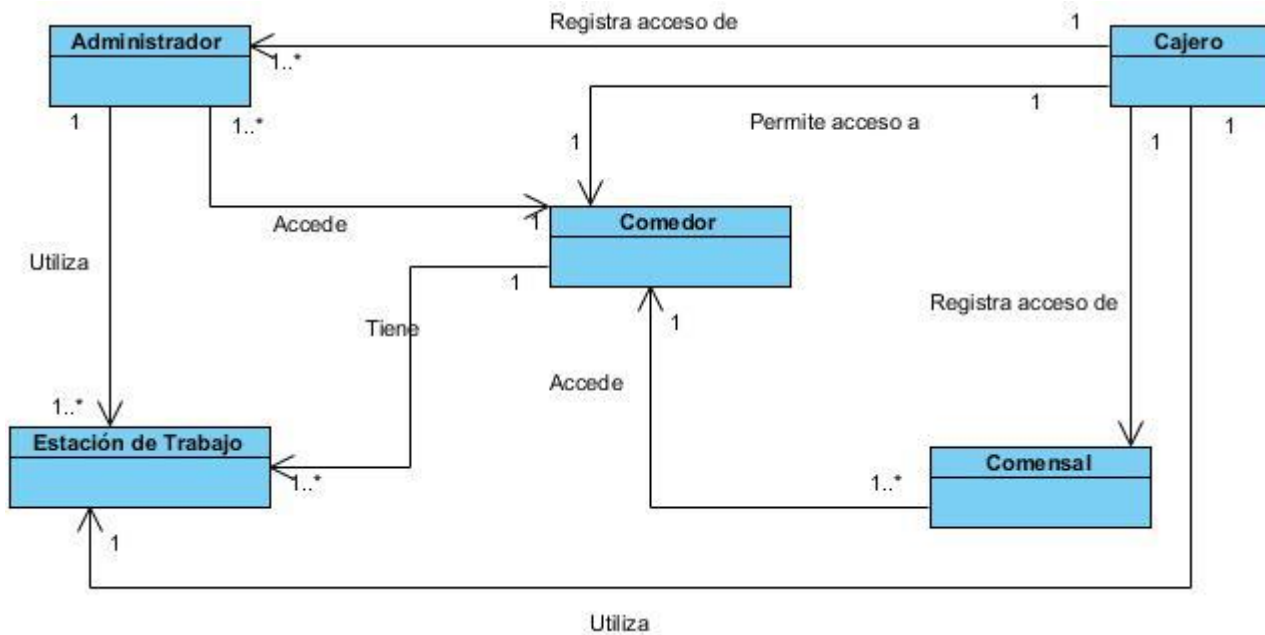
**Cajero:** persona encargada de controlar el acceso de estudiantes y trabajadores a los comedores mediante el sistema.

**Estación de trabajo:** estación de trabajo utilizada por el cajero y el administrador en el control de acceso.

**Comensal:** Persona interna o externa a la universidad que consume los servicios de esta.

**Comedor:** Local donde se encuentran las estaciones de trabajo y radica tanto el cajero como el administrador.

A continuación, se muestra el diagrama de clases del dominio donde se aprecia la relación entre las clases y la manera en que estas interactúan entre sí:



**Ilustración 6: Modelo de dominio.**

Comunicación entre las clases:

- La clase cajero comunica con las clases comensal y administrador cuando estas solicitan acceso a los comedores, que es concedido o no por el sistema manejado por el cajero.
- La clase cajero se comunica con la clase estación de trabajo cuando este para conceder o negar acceso hace uso del sistema de control de acceso alojado en la estación de trabajo.
- La clase cajero mantiene comunicación con la clase comedor puesto que este es el local al que él concede o niega el acceso.
- La clase comensal comunica con la clase comedor cuando este solicita el servicio de alimentación que se oferta en ese local.
- La clase administrador se comunica con la clase estación de trabajo cuando este registra la cantidad de raciones a ofertar en el día.
- La clase administrador comunica con la clase comedor cuando este solicita el servicio de alimentación que se oferta en ese local.
- La clase comedor comunica con la clase estación de trabajo pues es en este local donde se encuentran las estaciones de trabajo tanto del cajero como el administrador.



## 2.2 Propuesta de solución.

Como solución a la problemática descrita en la introducción del presente trabajo se propone la implementación de un sistema que permitirá informatizar el proceso de control de acceso físico a los comedores de la UCI. Dicho sistema mantendrá el control de las entradas de los comensales.

La interfaz del Sistema de Control de Acceso a comedores estará dividida en dos paneles principales: Panel superior el cual mostrará una barra de navegación donde se mostrarán las opciones: ver los datos de la cuenta del usuario y salir del sistema, además de contar con la funcionalidad: “Descargar Distribución”, que permitirá al usuario descargar la distribución previamente hecha por el Subsistema de Distribución. De igual manera contará con la opción de “Gestionar Evento”, permitiendo al usuario poder registrar, modificar y eliminar los eventos programados para el día. Por último, la funcionalidad “Buscar Comensal” encargada de controlar el acceso de los comensales buscándolos en la base de datos llenada por la opción “Descargar Distribución”.

Por último, el panel central mostrará la información de la funcionalidad que se ejecutará en el momento, así como el manejo de datos.

El sistema tiene dos tipos de roles, los cuales serán asignados a dos tipos de usuarios principales, que tendrán acceso al sistema: administradores y cajeros, estos son encargados de controlar el acceso a los comedores. El primer rol cuenta con permisos globales lo cual posibilita acceder y ejecutar cualquier funcionalidad o módulo del sistema. El segundo rol tiene los permisos de trabajar con los siguientes servicios:

- Módulo Buscar.

Tanto el administrador del sistema como el cajero poseen los permisos para registrar las entradas de los comensales.

Para acceder al sistema los usuarios autorizados deben autenticarse por LDAP. Una vez que haya ingresado al sistema se mostrará la ventana principal del mismo. Al introducir el número de solapín de un comensal, se podrá observar en el panel central los datos del usuario: nombre, tipo (interno o externo), usuario, solapín, género (masculino o femenino), área, categoría, cargo, provincia, municipio y teléfono, todo esto manejando desde una base de datos local. Comprobando los permisos disponibles para acceder al comedor. Una vez que el sistema

compruebe si el comensal tiene autorización para acceder se marcará como que ya ha recibido el servicio y no podrá repetirlo hasta que no comience un nuevo evento en el comedor. Si se deniega el acceso al comensal, aunque este cuente con la credencial, no podrá hacer uso de los servicios y el sistema notificará por donde tendría acceso para recibirlo.

El sistema registra el acceso de comensales verificando mediante el manejo de datos por LDAP. Se brinda la opción de “Gestionar Evento” y ver los detalles de los mismos. Para realizar una búsqueda de un comensal se hará mediante la opción de “Buscar”, ubicada en la parte central del panel superior, apareciendo un campo donde se introducirá el criterio de búsqueda.

## 2.3 Requisitos

La *IEEE, Standard Glossary of Software Engineering Terminology*, define un requisito como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Como parte del modelado se definen los requisitos. Los requisitos se clasifican en funcionales y no funcionales. Debido a la importancia que tienen los requerimientos deben ser especificados por escrito, claros y precisos.

### Técnicas de captura de requisitos

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. La ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma precisa.

A continuación, se presentan un grupo de técnicas que pueden ser usadas y que utilizó el autor de la investigación en la captura de los requerimientos.

**Entrevista:** es la técnica clásica para la captura de requisitos, se prepara con antelación la entrevista que después será aplicada al cliente. Se deben tener en cuenta cuestiones que se enfoquen sobre el cliente, los objetivos generales y los beneficios esperados.

La entrevista fue realizada a:

- Ing. Alexander Rodríguez Mompíé

Ingeniero en Ciencias Informáticas y Director de la Dirección de Informatización de la UCI (DIN), el cual estuvo directamente vinculado con el proceso de desarrollo del sistema de control de acceso que encuentra en explotación. Durante la entrevista se realizaron las siguientes preguntas:

1. ¿Por cada comensal que accede a los locales qué datos se deben reflejar en el sistema?
2. ¿Quién es el encargado de asignar las Raciones diarias de comida?
3. ¿Basándose en qué se concede o niega el acceso?
4. ¿Puede reconocer que el comensal que está accediendo a los comedores en ese momento tiene la autorización necesaria?

**Tormenta de ideas:** es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar.

Para realizar la tormenta de ideas se contó con la presencia de:

- Roberto Rodríguez Caballero (Autor de la investigación).
- Ing. Julio Cesar Espronceda Pérez (Tutor de la investigación).
- Ing. Yidier Romero Zaldívar (Tutor de la investigación).

Las técnicas de capturas de requisitos antes descritas permitieron arribar a un conjunto de requisitos funcionales y no funcionales que se presentan a continuación

### **2.3.1 Requisitos funcionales**

#### **Módulo Evento**

RF1- Registrar Evento (raciones, área, hora de inicio, hora de acabar, tipo).

RF2- Modificar Evento (raciones, área, hora de inicio, hora de acabar, tipo).

RF3- Mostrar Estado del Evento (raciones, área, hora de inicio, hora de acabar, tipo).

RF4-Descargar distribución.

### **Módulo Usuario**

RF5- Asignar permisos (usuario, administrador).

RF6- Autenticar Usuario (usuario, contraseña).

RF7-Adicionar Usuario (usuario, contraseña, rol).

### **Modulo Comedor**

RF8- Registrar comedor (complejo, puerta).

RF9- Modificar comedor (complejo, puerta).

RF10- Eliminar comedor

### **Módulo Búsqueda**

RF11- Buscar Comensal (código de barra).

## **2.3.2 Requisitos no funcionales**

Los requisitos no funcionales son las propiedades o cualidades que el sistema debe tener. Los requisitos no funcionales son aquellos que hacen que el sistema sea usable, rápido, confiable y agradable para los usuarios.

### **Usabilidad**

- El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

### **Fiabilidad**

- La información manejada por el sistema estará protegida de acceso no autorizado.

### **Rendimiento**

- Se tiene en cuenta que el producto se debe diseñar sobre una arquitectura cliente - servidor, los tiempos de respuestas del sistema deben ser rápidos: no debe de superar los 3 segundos.

### **Restricciones de diseño**

- El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.

### **Interfaz**

- El sistema debe contar con una interfaz fácil de usar, sencilla, intuitiva, permitiendo que los usuarios sean capaces de interactuar con la aplicación con facilidad.
- Será diseñada para adaptarse a la resolución de pantalla del usuario, utilizando colores como el gris y el azul.

### Seguridad

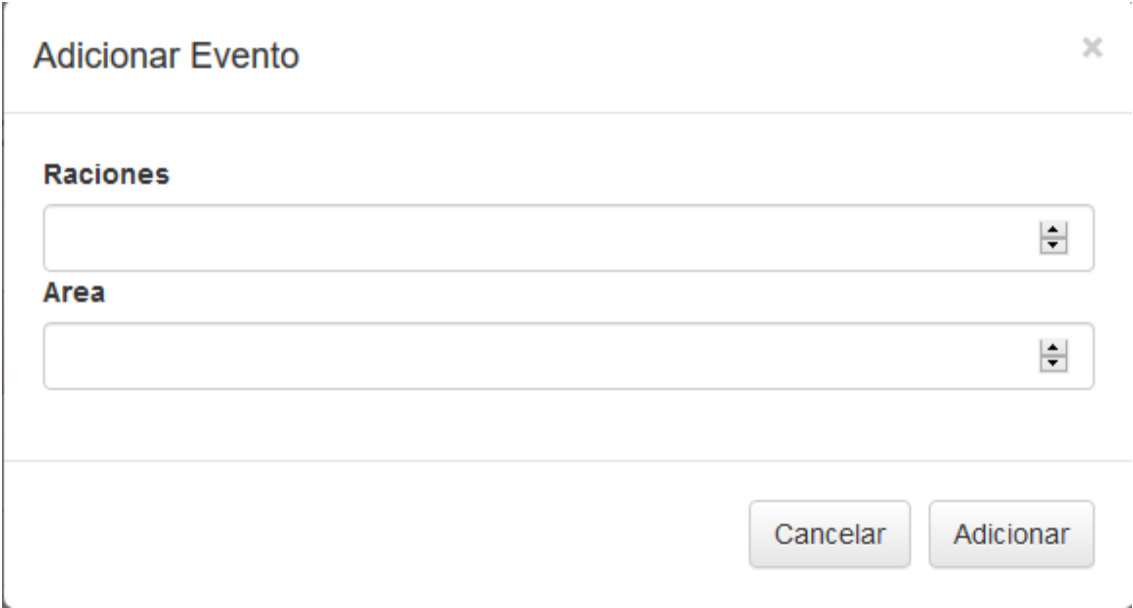
- Solo el usuario previamente autenticado podrá hacer uso del sistema.
- Confiabilidad: la información que se maneje en el sistema estará protegida de acceso no autorizado y divulgación, a partir de los diferentes roles de los usuarios que empleen el sistema.
- Integridad: la información manejada por el sistema será objeto de cuidadosa protección contra corrupción y estados inconsistentes, de igual manera el origen y autoridad de los datos. Se maneja el uso de la seguridad estableciendo roles. Se realizará el manejo de datos a través de *LDAP*.
- Disponibilidad: la información se encontrará disponible en todo momento para aquellos usuarios autorizados a acceder al sistema siempre y cuando la infraestructura tecnológica de la universidad lo permita.

### 2.4 Historia de Usuario.

A continuación, se presenta una muestra de las historias de usuario más relevantes para un estudio del resto de historias de usuario dirigirse al Anexo 1: Historias de Usuario.

**Tabla 1: Registrar Evento.**

<b>Número:</b> 1	<b>Nombre del requisito:</b> Registrar Evento
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días

<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días
<b>Descripción:</b>	
<b>1- Objetivo:</b>	
<i>Permitir introducir los datos de un evento.</i>	
<b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>	
Para registrar un evento debe cumplirse que:	
<i>- Estar autenticado en el sistema y tener permisos de administrador.</i>	
<b>3- Flujo de la acción a realizar:</b>	
<i>Cuando el usuario selecciona la opción Registrar Evento se visualiza una ventana donde se introducen los datos.</i>	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	
 <p>El prototipo de interfaz muestra una ventana modal titulada "Adicionar Evento" con un botón de cierre "x" en la esquina superior derecha. Dentro de la ventana, hay dos secciones de formularios: "Raciones" y "Area". Cada sección contiene un campo de entrada con un menú desplegable a la derecha. En la parte inferior de la ventana, hay dos botones: "Cancelar" y "Adicionar".</p>	

**Tabla 2: Modificar Evento.**

<b>Número:</b> 2	<b>Nombre del requisito:</b> Modificar Evento	
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 1ra	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días	
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días	
<b>Descripción:</b>  <b>1- Objetivo:</b>  <i>Permitir modificar los datos de un evento.</i>  <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>  Para modificar un evento debe cumplirse que:  <i>- Estar autenticado en el sistema y tener permisos de administrador.</i>  <i>-Debe haber al menos un evento registrado</i>  <b>3- Flujo de la acción a realizar:</b>  <i>Cuando el usuario selecciona la opción Modificar Evento se visualiza una ventana donde se introducen los datos en los campos a modificar.</i>		
<b>Observaciones:</b>		
<b>Prototipo de interfaz:</b>		

**Modificar Evento**
✕

---

**Raciones**

**Area**

**Tabla 3:Eliminar Evento.**

<b>Número: 3</b>	<b>Nombre del requisito:</b> Eliminar Evento
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días
<b>Descripción:</b>	
<p><b>1- Objetivo:</b></p> <p><i>Permitir Eliminar Evento.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b></p> <p>Para eliminar Evento debe cumplirse que:</p> <ul style="list-style-type: none"> <li>- <i>Estar autenticado en el sistema y tener permisos de administrador.</i></li> <li>-Debe haber al menos un horario registrado</li> </ul>	



### 3- Flujo de la acción a realizar:

*Cuando el usuario selecciona la opción Eliminar Evento se visualiza una ventana donde se notifica que se va a eliminar un evento con las opciones de aceptar y cancelar donde el usuario selecciona la acción deseada.*

### Observaciones:

### Prototipo de interfaz:

Eliminar Evento ×

---

Eliminar Evento

---

Cancelar

Eliminar

## 2.5 Arquitectura

La arquitectura de *software* se relaciona con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales. Estructura los componentes de un sistema de software, sus interrelaciones, los principios y directrices que rigen su diseño y evolución en el tiempo. Al definir la arquitectura de software de un sistema se establecen las responsabilidades arquitectónicas y los fundamentos para que los integrantes del equipo de desarrollo de software, trabajen en una línea común que permita alcanzar los objetivos del sistema y satisfacer las necesidades del cliente.

### 2.5.1 Patrones de diseño y estilo de arquitectura

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Facilitan la reutilización de arquitecturas y diseños de *software* exitosos. Los patrones de arquitectura son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de *software*. Dan una descripción de los elementos y el tipo de relación que tienen junto con un grupo de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor.

#### Patrón modelo vista controlador

El modelo vista controlador por sus siglas en inglés MVC es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo mayor independencia, mantenimiento y reutilización.

Modelo: representa los datos del programa y controla todas sus transformaciones.

Vista: genera la presentación visual de los datos representados por el modelo y muestra los datos al usuario.

Controlador: maneja las entradas del usuario, actuando sobre los datos representados por el modelo.

#### Ventajas

1. Se divide la lógica de negocio del diseño, haciendo tu proyecto más escalable.
2. Facilita el uso de URL amigables, importantes para el SEO (Posicionamiento web), la mayoría de *frameworks* MVC lo controlan.
3. Muchos *frameworks* MVC ya incluyen librerías de Javascript como *Jquery*, lo que facilitará validar formularios en el cliente y en el servidor.
4. Puedes utilizar abstracción de datos, facilitando la realización de consultas a la base de datos.

5. Poco a poco el desarrollo web se orienta a lo que se denomina "*Agile Web Development*" (Desarrollo ágil de aplicaciones web), con frameworks como *Ruby on Rails* que ayudan a crear proyectos de calidad y en corto tiempo. Existen varios *frameworks* en PHP e incluso *ASP .NET* que en su nueva versión ya contempla el MVC con *Visual C#*.
6. Un *Framework MVC* te ayuda a controlar los recursos del servidor, evitando *Bugs* que puedan repercutir en el rendimiento, por ejemplo, muchas veces olvidamos cerrar conexiones a la base de datos, sobrecargando el servidor.

Se hace uso del *MVC* por las ventajas y características que el propio posee.

## **CONCLUSION PARCIAL**

- A partir de las técnicas de captura de requisitos entrevista y tormenta de ideas, se obtuvieron los requerimientos funcionales y no funcionales de propuesta de solución.
- Se elige el patrón de diseño modelo vista controlador que permitió separar los datos, la lógica de negocio y las interfaces de usuario.
- El uso de Historias de usuario permitió representar los requisitos funcionales antes expuestos.

## CAPÍTULO 3: EVALUACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA

### Introducción:

En el presente capítulo se definen los diagramas de componentes y despliegue, los cuales tienen un papel fundamental en la etapa de construcción del sistema. Como cierre del capítulo se muestra el resultado de las pruebas realizadas a la aplicación desarrollada.

### 3.1 Tareas de ingeniería.

A continuación, se muestran las tareas de ingeniería correspondientes a las historias de usuario mostradas en el capítulo anterior, para un mayor estudio de las mismas dirigirse al Anexo 3: Tareas de ingeniería.

**Tabla 4 : Registrar evento.**

Tarea de ingeniería	
Numero de Tarea:1	Historia de Usuario: No.1 Registrar evento
Nombre Tarea: Desarrollar el registro de eventos	
Tipo de Tarea: Desarrollo	Iteración: 1
Fecha Inicio: 14 de diciembre 2017	Fecha Fin:21 de diciembre 2017
Programador Responsable: Roberto Rodríguez Caballero	
Descripción:  <b>1- Objetivo:</b>  <i>Permitir Registrar un Evento.</i>  <b>2- Acciones para lograr el objetivo:</b>  -Desarrollar los formularios y vistas correspondientes con la funcionalidad.  - Implementar la clase controladora para la funcionalidad Registrar evento.	

**Tabla 5: Modificar evento.**

Tarea de ingeniería	
Numero de Tarea:2	Historia de Usuario: No.2 Modificar evento
Nombre Tarea: Desarrollar Modificar de eventos	
Tipo de Tarea: Desarrollo	Iteración: 1
Fecha Inicio: 21 de diciembre 2017	Fecha Fin:30 de diciembre 2017
Programador Responsable: Roberto Rodríguez Caballero	
Descripción: <b>1- Objetivo:</b> <i>Permitir Modificar un Evento.</i> <b>2- Acciones para lograr el objetivo:</b> -Desarrollar los formularios y vistas correspondientes con la funcionalidad. - Implementar la clase controladora para la funcionalidad Modificar evento.	

**Tabla 6: Eliminar evento.**

Tarea de ingeniería	
Numero de Tarea:3	Historia de Usuario: No.3 Eliminar evento
Nombre Tarea: Desarrollar el registro de eventos	
Tipo de Tarea: Desarrollo	Iteración: 1
Fecha Inicio: 4 de enero 2018	Fecha Fin:11 de diciembre 2018
Programador Responsable: Roberto Rodríguez Caballero	
Descripción:	

### 1- Objetivo:

*Permitir Eliminar un Evento.*

### 2- Acciones para lograr el objetivo:

- Desarrollar los formularios y vistas correspondientes con la funcionalidad.
- Implementar la clase controladora para la funcionalidad Eliminar evento.

## 3.2 Estándares de codificación.

**Definición de las clases:** A continuación, se muestra un ejemplo de la definición de una de las clases del sistema, en este caso Evento.

```
@Table(name = "Evento")
public class Evento implements Serializable
```

### Ilustración 7 :Definición de clases

**Estructuras de control:** En siguiente fragmento de código se aprecia el uso de la estructura de control *if*.

```
</li>
<%if (listado_comedor.isEmpty() != true) {%>
<li><%= listado_comedor.get(0).getPuerta() %></li>
<%}%>
```

### Ilustración 8 : Estructura de control

**Asignación de variables:** A continuación, se muestra de cómo fueron asignadas las variables.

```
Long id =Long.parseLong(request.getParameter("mid"));
String raciones = request.getParameter("modraciones");
String area=request.getParameter("modarea");
```

### 3.2 Diagrama de componente.

Los diagramas de componentes muestran los elementos de diseño de un sistema de *software*. Permite visualizar con facilidad la estructura general del sistema y el comportamiento del servicio que estos proporcionan y utilizan a través de las interfaces. Muestran tanto los componentes de *software* como las relaciones lógicas entre ellos en un sistema. Representan todos los tipos de elementos del *software* implicados en la fabricación de aplicaciones informáticas.

A continuación, se presenta el diagrama de componente del sistema:

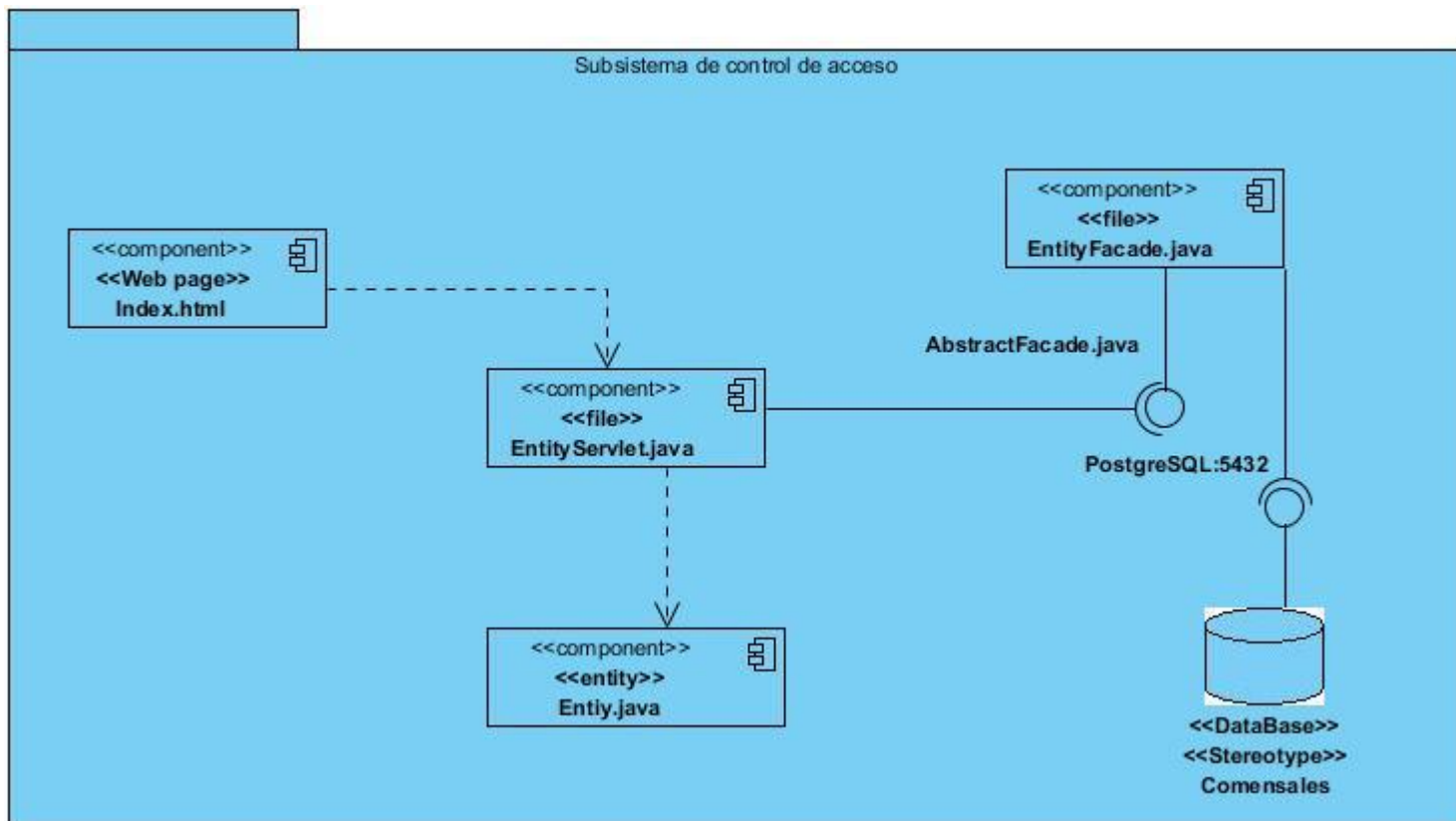
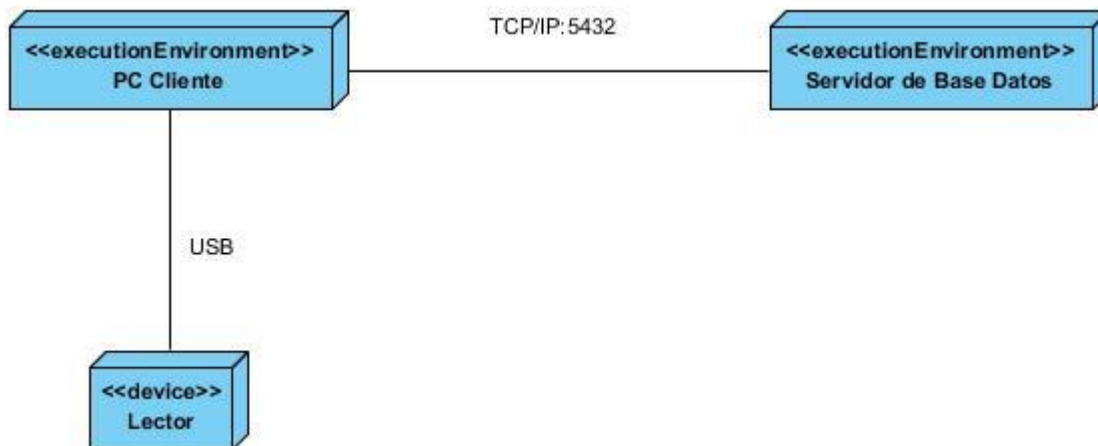


Ilustración 9: Diagrama de componente.

### 3.3 Diagrama de despliegue

El diagrama 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. Es una colección de nodos donde uno representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.

A continuación, se presenta el diagrama de despliegue que se propone para el sistema:



**Ilustración 10 : Diagrama de despliegue del sistema.**

**Servidor Base de Datos:** este nodo representa el servidor de base de datos, que provee servicios de base de datos a las PC Cliente.

**PC Cliente:** este nodo representa todas aquellas estaciones de trabajo asignadas a los cajeros, donde se incluirá el sistema para la administración de los accesos físicos a los comedores.

**Lector:** este nodo representa al lector de código de barra usado para capturar los datos de la credencial de los comensales.

**Descripción del tipo de comunicación:** La comunicación entre el nodo que representa la Pc Cliente y el nodo Servidor Base de Datos se realiza mediante el protocolo TCP/IP por el puerto 5432. Los nodos que representan La PC cliente y el lector se comunican mediante USB.

### 3.4 Pruebas

Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Es importante destacar que las pruebas reducen la probabilidad de la presencia de los defectos ocultos en el *software*, incluso si no existe ningún defecto, esto no será garantía de su corrección. Cualquier producto de ingeniería puede probarse de una de estas dos formas: Conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y al mismo tiempo buscando errores en cada función. Conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. El primer enfoque de



prueba se denomina prueba de caja negra y el segundo, prueba de caja blanca. Al conocer la función específica para la que fue diseñado el sistema desarrollado, se decide aplicar las pruebas de caja negra.

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

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. Esta intenta encontrar errores de las siguientes categorías: Funciones incorrectas o ausentes. Errores de interfaz. Errores en estructuras de datos o en accesos a bases de datos externas. Errores de rendimiento. Errores de inicialización y de terminación. Hacen referencia a las pruebas que se llevan a cabo sobre la interfaz del software. Las pruebas de caja negra se basan en casos de prueba. Los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto.

Para la realización de las pruebas existen varias técnicas:

La técnica que se propone para realizar los casos de pruebas es la variante de particiones equivalentes, se basa en dividir el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*. la cual consta de dos pasos fundamentales:

- Identificación de las clases de equivalencia, es decir, los conjuntos de estados válidos o no válidos para condiciones de entrada.
- Identificar los casos de pruebas.

En el primer paso, las clases de equivalencia son identificadas tomando cada condición de entrada (generalmente una oración o una frase en la especificación) y repartiéndola en dos o más grupos, cada combinación será un escenario de las historias de usuario. En el segundo paso se identifican las variables y las clases de equivalencia para la confección de los casos de prueba.

Los Casos de Prueba que se utilizarán para las pruebas del sistema están diseñados para verificar que se satisfacen los requerimientos del usuario, tal y como se describe en las especificaciones de los Casos de Usos. A continuación, se muestran los Casos de Prueba para la Historias de Usuario (HU) descritas en el capítulo anterior, para estudios del resto de casos de prueba dirigirse al Anexo 2: Casos de prueba.

**Tabla 7 :Descripción de la HU Registrar evento.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Registrar Evento.	Registrar todos los datos de un evento para añadirlo en el sistema.

**Tabla 8: Variables de la HU Registrar evento.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Raciones	Campo a llenar	No	Campo donde se introduce la cantidad de raciones para el evento.
Área	Campo a llenar	No	Campo donde se introduce en área del evento .
Hora de inicio	Campo a llenar	No	Campo donde se introduce la hora en que empieza el evento.
Hora de fin	Campo a llenar	No	Campo donde se introduce la hora en que acaba el evento.
Tipo	Campo a llenar	No	Campo donde se especifica el tipo de evento.
Registrar	Botón	No	Opción de Registrar.
Cancelar	Botón	No	Opción de Cancelar el Registro.

**Tabla 9: Caso de prueba HU Registrar evento.**

Id del Escenario	Escenario	Raciones	Área	Hora de inicio	Hora de terminar	Registrar	Cancelar	Resultado de la Prueba	Respuesta del sistema
------------------	-----------	----------	------	----------------	------------------	-----------	----------	------------------------	-----------------------

<b>EC1</b>	Selecciona la opción Registrar Evento.	V	V	V	V	N/A	N/A	Satisfactoria	Se muestra un formulario con los campos a llenar por el usuario.
<b>EC2</b>	Introduce los datos y selecciona la opción Registrar.	V	V	V	V	V	N/A		Se muestra los detalles del nuevo Evento . - Raciones . -Local Y las opciones de editar y eliminar.
<b>EC3</b>	Campo incompleto .	V	V	V	V	V	N/A		Se muestra un mensaje, "Rellene este campo".

**Tabla 10: Descripción de la HU modificar evento.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Modificar Evento.	Modificar todos o algún dato de un evento y agregarlo en el sistema.

**Tabla 11: Variables de la HU Modificar evento.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Raciones	Campo a llenar	No	Campo donde se introduce la cantidad de raciones para el evento.
Área	Campo a llenar	No	Campo donde se introduce el área del evento.
Hora de inicio	Campo a llenar	No	Campo donde se introduce la hora de inicio del evento.
Hora de fin	Campo a llenar	No	Campo donde se introduce la hora de finalizar el evento.
Tipo	Campo a llenar	No	Campo donde se especifica el tipo de evento.
Modificar	Botón	No	Opción de Modificar.
Cancelar	Botón	No	Opción de Cancelar la Modificación.

**Tabla 12: Caso de prueba HU Modificar evento.**

Id del Escenario	Escenario	Raciones	Área	Hora de inicio	Hora de fin	Tipo	Modificar	Cancelar	Resultado de la Prueba	Respuesta del sistema
<b>EC1</b>	Selecciona la opción Modificar Evento.	V	V	V	V	V	N/A	N/A	Satisfactoria	Se muestra un formulario con los campos a llenar por el usuario.
<b>EC2</b>	Introduce los datos y selecciona la opción Modificar.	V	V	V	V	V	V	N/A		Se muestra los detalles del nuevo Evento . – Raciones . -Local Y las opciones de editar y eliminar.
<b>EC3</b>	Campo incompleto	V	V	V	V	V	V	N/A		Se muestra un

										mensaje, "Rellene este campo".
--	--	--	--	--	--	--	--	--	--	---

**Tabla 13: Descripción de la HU Eliminar evento.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Eliminar Evento	Eliminar uno o todos los eventos planificados en el día.

**Tabla 14: Variables de la HU Eliminar evento.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Eliminar	Botón	No	Opción Eliminar.
Aceptar	Botón	No	Opción de Aceptar.
Cancelar	Botón	No	Opción de Cancelar

**Tabla 15: Caso de prueba HU Eliminar evento.**

Id del Escenario	Escenario	Eliminar	Aceptar	Cancelar	Resultado de la Prueba	Respuesta del sistema
EC1	Selecciona la opción Eliminar	V	N/A	N/A	Satisfactoria	Muestra un mensaje ¿Estás seguro que desea eliminar este elemento?.
EC2	Presiona el botón Aceptar.	N/A	V	V		Se elimina el evento.
EC3	Presiona	N/A	V	V		El sistema regresa a

	el botón Cancelar					la página principal mostrando el listado de evento.
--	-------------------	--	--	--	--	---

### Resultados de las pruebas.

Para evaluar la solución desarrollada se realizaron tres iteraciones de pruebas. Se abarcaron los métodos y técnicas de pruebas expuestas anteriormente en cada una de las iteraciones, arrojando resultados que demuestran la calidad de la aplicación desarrollada. En la primera iteración se detectaron catorce No Conformidades, de las cuales nueve eran de errores de interfaz, cuatro de funcionalidad, uno de ortografía. Una vez resueltas las no conformidades, se realizó la segunda iteración detectándose cuatro NC, de las cuales dos era de funcionalidad y dos de interfaz. Al realizarse la tercera iteración no se encontraron NC, dando como resultado que el sistema se encontraba listo para ser desplegado, lo cual no garantiza la no existencia de nuevos errores como se muestra en la siguiente gráfica.

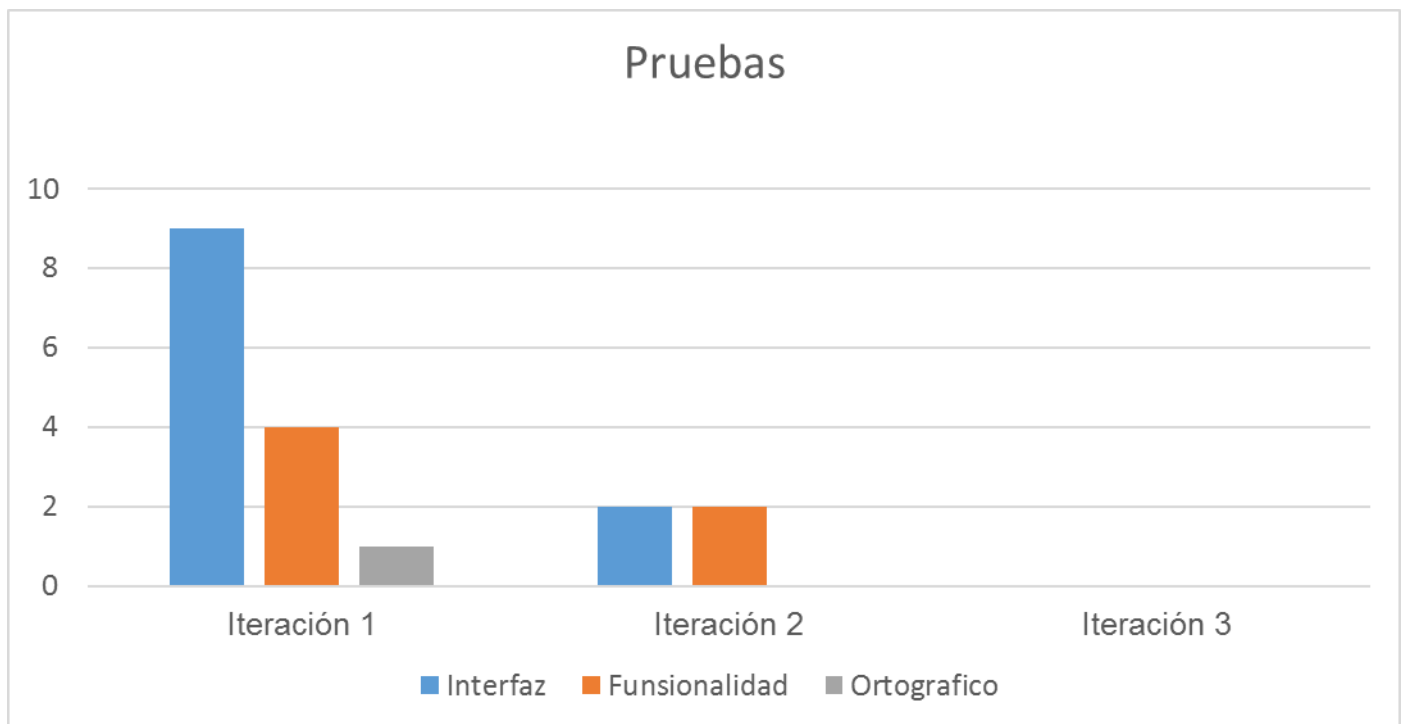


Ilustración 11 :Resultados de pruebas.

### **3.5 Conclusiones Parciales.**

- La correcta selección y ubicación de los componentes facilitó la implementación del *software*, haciendo rápido el desarrollo del mismo y permitiendo un mejor entendimiento de los elementos a implementar y la estructura de estos, a través de la realización de los diagramas de componentes y despliegue.
- Los métodos de pruebas ejecutados validaron el correcto funcionamiento de la aplicación, cumpliendo con los requisitos especificados y evidenciaron que el software está listo para ser implantado.



## **Conclusiones**

- El análisis de los diferentes sistemas dedicados al control de acceso físico y lógico arribó a que ninguna de las soluciones satisface las nuevas necesidades y restricciones entorno al acceso a comedores en la UCI y se logró identificar las características y funcionalidades más comunes de estos sistemas para una posible propuesta de solución.
- El uso de las herramientas, tecnologías y lenguajes de programación que soportaron el proceso de desarrollo guiado por la metodología AUP-UCI, permitieron obtener un producto con la calidad requerida que cumple con los requisitos funcionales y no funcionales definidos en la investigación.
- La realización de pruebas de caja negra y el análisis de los resultados de las mismas permitieron determinar y erradicar las deficiencias encontradas, garantizando la solidez del sistema.

## **RECOMENDACIONES**

- Realizar el despliegue del sistema propuesto a todos los comedores de la Universidad.
- Utilizar el sistema como sustituto del subsistema actual.

## REFERENCIAS BIBLIOGRÁFICAS

1. Almagro, C. U. (2011). *Lenguajes de Programación*.
2. Gauchat, S. D. (s.f.). *El gran libro de HTML, CSS3 Y JAVASCRIPT*.
3. *GetBootstrap.com*. (s.f.). Obtenido de GetBootstrap.com: <https://getbootstrap.com/>
4. Gonçalves, E. (2008). *Desenvolviendo aplicações Web com NetBeans IDE*.
5. SANABRIA, E. C. (2002). Introducción a Java. *JAVA BASICO*, (pág. 6).
6. SCALA. (10 de 2003). *smartcardalliance.org*. Obtenido de smartcardalliance.org: [www.smartcardalliance.org](http://www.smartcardalliance.org)
7. Timothy A. Howes, M. C. (2003). *Understanding and Deploying LDAP Directory Services*.
8. UCI. (9 de enero de 2018). *UCI*. Obtenido de [www.uci.cu](http://www.uci.cu)
9. Yasmani Freixas Ramírez, V. N. (2015). *Sistema para el control de acceso a los laboratorios del Centro de Tecnologías para la Formación*.
10. Sistemas de control de acceso y software de control de acceso HID. [En línea] SmartCardSystems S.A, 2013 de enero de 2013. <http://www.scssa.com.ar/control-deacceso.htm>.
11. Menezes, Alfred J. Hand book of applied cryptography.[En línea] Octubre de 1996. <http://cacr.uwaterloo.ca/hac/>.
12. Mora, Sergio Lujan. Programación de aplicaciones web: historia, principios básicos y clientes web .2002.
13. Culebro, y otros. Software libre vs software propietario. México: s.n., 2006.
14. Ian. Somerville Ingeniería del Software. Madrid: s.n., 2005.
15. Carmona, Rhadamés. El Enfoque Orientado a Objetos. 2004.
16. Solís, Camilo J., Figueroa, Roberth G. y Cabrera, Armando A. METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación: s.n.
17. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. S.l.: s/n, 2000.
18. Metodologías Ágiles. (Proceso Unificado Ágil). Bolivia: s.n.
19. Modeller, Umbrello UML. Manual de Umbrello UML Modeller. 2003.
20. González, José Enrique y Cornejo. ¿Qué es UML? 2005.
21. Allweyer, Thomas. Introduction to the Standard for Business Process Modeling. 2010. BPMN 2.0.
22. Aranda, Vicente Trigo. Historia y evolución de los lenguajes de programación. 2001.
23. Veracruz, Instituto Tecnológico de. Perspectiva histórica del Internet. 2007.
24. Álvarez, M. Introducción a PHP 5. 2004.
25. Domínguez, M. Dorado. Bases de datos en el cliente con JavaScript DB. Madrid: Iberprensa, 2005., 2005.
26. Markiewicz, Carlo, Marcus, Eduardo y Lucena, J.P. El Desarrollo del Framework Orientado al Objeto. 2001.
27. Gutiérrez, Javier. ¿Qué es un framework web? 2010.
28. León, Eduardo. Tutorial Visual Paradigm for UML. 2000.

29. Domínguez Dorado, M. NetBeans IDE 4.1. La alternativa a Eclipse. Madrid: Iberprensa, 2005.
30. Sánchez, Jorge. Servidores de Aplicaciones Web. 2011.
31. Foundation. The Apache Software. Apache License and Distribution.2012.
32. Bertino, E. A. y MARTINO, L. Sistemas de bases de datos orientadas a objetos. s.l. 1995., 1995.
33. Pecos, Daniel. PostgreSQL vs. MySQL.2008.
34. Battersby, Jeffery. Web browser round up. 2005.
35. García, José Manuel. LA ÉTICA COMO ASIGNATURA EN LOS ESTUDIOS DE. [En línea] 15 de noviembre de 1999. [Citado el: 05 de febrero de 2015.] <http://ditec.um.es/~jmgarcia/papers/etica.pdf>.
36. Guadalupe. LAS TIC EN TRABAJO SOCIAL. [En línea] 19 de marzo de 2012. [Citado el: 05 de febrero de 2015.] <http://ticyts.blogspot.com/2012/03/evolucion-de-lastic-opportunidades-y.html>.
37. EcuRed. ecured. [En línea] [Citado el: 7 de febrero de 2015.] [http://www.ecured.cu/index.php/Sistemas\\_de\\_control\\_de\\_acceso](http://www.ecured.cu/index.php/Sistemas_de_control_de_acceso).
38. Escribano, Gerardo Fernández. Introducción a Extreme Programming .9-122002.
39. El modelo scrum. S.l.: NST-0010 Rev. 0.1 Navegapolis.net.
40. Recio, Uuneikys y Montero, Osbel. Desarrollo de la Aplicación del Portal Web del CICPC 2010.
41. Arquero Sistema corporativo. Software de control y seguridad. Control de acceso y control de horario. [En línea] 15 de enero de 2013. <http://www.scispain.com/root.php?modulo=controlAcceso>.
42. Miranda, Aquiles Pérez. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. Sistema de control de acceso automatizado para los laboratorios de la Universidad de las Ciencias Informáticas. La Habana: s.n., 2013.
43. El Nuevo Estándar en Control de Acceso. SE®, Plataforma iCLASS. EE.UU.: 2013 HID Global Corporation/ASSA ABLOY AB., 2013.

## ANEXOS

### Anexo 1: Historias de usuario

**Tabla 16: Mostrar estado de evento.**

<b>Número:</b> 5	<b>Nombre del requisito:</b> Mostrar estado de evento
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 15 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días
<b>Descripción:</b>  <b>1- Objetivo:</b>  <i>Permitir mostrar el estado actual de los eventos registrados.</i>  <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>  Para mostrar el estado de un evento debe cumplirse que:  - <i>Estar autenticado en el sistema.</i>  - Debe haber al menos un evento registrado  <b>3- Flujo de la acción a realizar:</b>  <i>Cuando el usuario selecciona la opción Registrar Evento se visualiza una ventana donde se introducen los datos de un evento que se visualizaran en la pantalla principal.</i>	
<b>Observaciones:</b>	

**Prototipo de interfaz:**

**Tabla 17: Mostrar detalles de evento.**

<b>Número:</b> 6	<b>Nombre del requisito:</b> Mostrar detalles de evento
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días
<b>Descripción:</b>  <b>1- Objetivo:</b>  <i>Permitir mostrar los detalles de un evento.</i>  <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>  Para mostrar detalles de un evento debe cumplirse que:  - <i>Estar autenticado en el sistema.</i>  - Debe haber al menos un evento registrado  <b>3- Flujo de la acción a realizar:</b>  <i>Cuando el usuario selecciona la opción mostrar detalles se visualizan los detalles del evento seleccionado.</i>	
<b>Observaciones:</b>	

**Prototipo de interfaz:**

**Tabla 18: Autenticar Usuario.**

<b>Número:</b> 7	<b>Nombre del requisito:</b> Autenticar Usuario
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días
<b>Descripción:</b>  <b>1- Objetivo:</b>  <i>Permitir autenticar a un usuario en el sistema.</i>  <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>  Para autenticar un usuario debe cumplirse que:  <i>- Debe ser administrador de comedor o cajero de comedor.</i>  <b>3- Flujo de la acción a realizar:</b>  <i>Cuando el usuario ejecuta el sistema se visualiza una ventana donde el usuario debe introducir su usuario y contraseña.</i>	
<b>Observaciones:</b>	

**Prototipo de interfaz:**

The image shows a login form prototype on a light gray background. At the top, the text "Inicie Sesión" is displayed in a large, dark font. Below this, there are two white input fields: the first is labeled "Usuario" and the second is labeled "Password". Underneath the password field is a checkbox followed by the text "Recuerdame". At the bottom of the form is a blue button with the text "Entrar" in white.

**Tabla 19: Registrar comedor.**

<b>Número:</b> 11	<b>Nombre del requisito:</b> Registrar comedor	
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 2da	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días	
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días	
<b>Descripción:</b>		
<b>1- Objetivo:</b>		
<i>Permitir Registrar un comedor.</i>		



**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para Registrar un comedor debe cumplirse que:

- *Estar autenticado en el sistema y tener permisos de administrador.*

**3- Flujo de la acción a realizar:**

*Cuando el usuario selecciona la opción Registrar comedor se visualiza una ventana donde se introducen los datos.*

**Observaciones:**

**Prototipo de interfaz:**

Adicionar Comedor [X]

**Puerta no.1**

**Complejo**

Cancelar Adicionar

**Tabla 17: Modificar comedor.**

<b>Número:</b> 12	<b>Nombre del requisito:</b> Modificar comedor
<b>Programador:</b> Roberto Rodríguez	<b>Iteración Asignada:</b> 2da

Caballero	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b></p> <p><i>Permitir modificar los datos de un comedor.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b></p> <p>Para modificar un evento debe cumplirse que:</p> <ul style="list-style-type: none"> <li>- <i>Estar autenticado en el sistema y tener permisos de administrador.</i></li> <li>- <i>Debe haber al menos un comedor registrado.</i></li> </ul> <p><b>3- Flujo de la acción a realizar:</b></p> <p><i>Cuando el usuario selecciona la opción Modificar comedor se visualiza una ventana donde se introducen los datos en los campos a modificar.</i></p>	
<b>Observaciones:</b>	
<p><b>Prototipo de interfaz:</b></p>	

Adicionar Comedor
✕

---

**Puerta no.1**

**Complejo**

---

Cancelar
Adicionar

**Tabla 18: Buscar comensal.**

<b>Número:</b> 14	<b>Nombre del requisito:</b> Buscar comensal		
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 2da		
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días		
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días		
<b>Descripción:</b>			
<b>1- Objetivo:</b>			
<i>Permitir buscar un comensal en la base de datos.</i>			
<b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>			
Para Buscar un comensal debe cumplirse que:			

- *Estar autenticado en el sistema.*

**3- Flujo de la acción a realizar:**

*Cuando el usuario selecciona la opción Buscar comensal se visualiza los datos del comensal del que fueron introducidos los datos de búsqueda.*

**Observaciones:**

**Prototipo de interfaz:**

**Tabla 19: Asignar Permisos.**

<b>Número:</b> 15	<b>Nombre del requisito:</b> Asignar Permisos
<b>Programador:</b> Roberto Rodríguez Caballero	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 10 días
<b>Descripción:</b>	
<b>1- Objetivo:</b>	
<i>Permitir establecer roles a los usuarios del sistema.</i>	
<b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>	
Para asignar permisos debe cumplirse que:	
<i>- Estar autenticado en el sistema y tener permisos de administrador.</i>	

### 3- Flujo de la acción a realizar:

Cuando el usuario selecciona la opción *Asignar Permisos* se visualiza una ventana donde se muestran los usuarios del sistema ,el usuario selecciona a quien asignarle los permisos.

### Observaciones:

### Prototipo de interfaz:

Adicionar Usuario

Usuario

Contraseña

Rellene este campo.

Rol

Cancelar Adicionar

### Anexo 2: Tareas de ingeniería.

#### Tabla 20 : Registrar comedor.

Tarea de ingeniería	
Numero de Tarea:4	Historia de Usuario: No.4 Registrar comedor
Nombre Tarea: Desarrollar el registro de comedor	

Tipo de Tarea: Desarrollo	Iteración: 1
Fecha Inicio: 14 de enero 2018	Fecha Fin:20 de enero 2018
Programador Responsable: Roberto Rodríguez Caballero	
<p>Descripción:</p> <p><b>1- Objetivo:</b></p> <p><i>Permitir Registrar un comedor.</i></p> <p><b>2- Acciones para lograr el objetivo:</b></p> <p>-El sistema debe mostrar un formulario con los campos a llenar por el usuario.</p> <p>- El sistema agrega el comedor con los datos introducidos.</p>	

**Tabla 21: Modificar comedor**

<b>Tarea de ingeniería</b>	
Numero de Tarea:5	Historia de Usuario: No.5 Modificar comedor
Nombre Tarea: Desarrollar el registro de comedor	
Tipo de Tarea: Desarrollo	Iteración: 1
Fecha Inicio: 14 de enero 2018	Fecha Fin:20 de enero 2018
Programador Responsable: Roberto Rodríguez Caballero	
<p>Descripción:</p> <p><b>1- Objetivo:</b></p> <p><i>Permitir Modificar un comedor.</i></p> <p><b>2- Acciones para lograr el objetivo:</b></p> <p>-El sistema muestra un formulario con los campos a llenar.</p>	

-El sistema modifica el comedor con los nuevos datos.

**Tabla 22: Mostrar detalles de evento.**

Tarea de ingeniería	
Numero de Tarea:6	Historia de Usuario: No.6 Mostrar detalles de evento.
Nombre Tarea: Desarrollar Mostrar detalles de evento.	
Tipo de Tarea: Desarrollo	Iteración: 2
Fecha Inicio: 21 de enero 2018	Fecha Fin:30 de enero 2018
Programador Responsable: Roberto Rodríguez Caballero	
<p>Descripción:</p> <p><b>1- Objetivo:</b></p> <p><i>Permitir mostrar los detalles de los eventos programados.</i></p> <p><b>2- Acciones para lograr el objetivo:</b></p> <p>-El sistema muestra una tabla con los detalles del evento.</p>	

**Tabla 23: Buscar comensal**

Tarea de ingeniería	
Numero de Tarea:7	Historia de Usuario: No.7 Buscar comensal
Nombre Tarea: Desarrollar buscador.	
Tipo de Tarea: Desarrollo	Iteración: 2
Fecha Inicio: 3 de febrero 2018	Fecha Fin:15 de febrero 2018

Programador Responsable: Roberto Rodríguez Caballero
<p>Descripción:</p> <p><b>1- Objetivo:</b></p> <p><i>Permitir buscar a un comensal dado el solapín.</i></p> <p><b>2- Acciones para lograr el objetivo:</b></p> <p>-El sistema muestra una barra de navegación con la opción de buscar.</p> <p>-El sistema muestra los datos del comensal encontrado.</p>

**Tabla 24 :Autenticar usuario.**

Tarea de ingeniería	
Numero de Tarea:8	Historia de Usuario: No.8 Autenticar usuario.
Nombre Tarea: Desarrollar autenticar usuario.	
Tipo de Tarea: Desarrollo	Iteración: 2
Fecha Inicio: 15 de febrero 2018	Fecha Fin:20 de febrero 2018
Programador Responsable: Roberto Rodríguez Caballero	
<p>Descripción:</p> <p><b>1- Objetivo:</b></p> <p><i>Permitir a un usuario acceder al sistema mediante usuario y contraseña.</i></p> <p><b>2- Acciones para lograr el objetivo:</b></p> <p>-El sistema muestra un formulario para llenar con los campos de usuario y contraseña.</p> <p>-El sistema niega o concede el acceso según los parámetros llenados por el usuario.</p>	



**Tabla 25 :Asignar permiso.**

Tarea de ingeniería	
Numero de Tarea:9	Historia de Usuario: No.9 Asignar permisos
Nombre Tarea: Desarrollar asignar permisos.	
Tipo de Tarea: Desarrollo	Iteración: 2
Fecha Inicio: 15 de febrero 2018	Fecha Fin:20 de febrero 2018
Programador Responsable: Roberto Rodríguez Caballero	
<p>Descripción:</p> <p><b>1- Objetivo:</b></p> <p><i>Permitir a un usuario conceder permisos de administración o usuario.</i></p> <p><b>2- Acciones para lograr el objetivo:</b></p> <ul style="list-style-type: none"> <li>-El sistema muestra un formulario para llenar por el usuario.</li> <li>-El sistema da privilegios a los usuarios dependiendo de los roles asignados.</li> </ul>	

**Tabla 26 :Mostrar estado de evento.**

Tarea de ingeniería	
Numero de Tarea:10	Historia de Usuario: No.10 Mostrar estado de evento.
Nombre Tarea: Desarrollar mostrar estado de evento.	
Tipo de Tarea: Desarrollo	Iteración: 2
Fecha Inicio: 15 de abril 2018	Fecha Fin:30 de abril 2018

Programador Responsable: Roberto Rodríguez Caballero
<p>Descripción:</p> <p><b>1- Objetivo:</b></p> <p><i>Permitir a un usuario visualizar el estado actual de los eventos.</i></p> <p><b>2- Acciones para lograr el objetivo:</b></p> <p>-El sistema muestra una tabla con los datos del estado actual de los eventos.</p>

### Anexo 3: Casos de prueba.

**Tabla 27: Descripción de la HU Autenticar usuario.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Autenticar Usuario	Permitir que los usuarios accedan al sistema mediante su usuario y contraseña

**Tabla 28: Variables de la HU Autenticar usuario.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Usuario	Campo a llenar	No	Campo donde se introduce el usuario.
Contraseña	Campo a llenar	No	Campo donde se introduce la contraseña.
Entrar	Botón	No	Opción de entrar al sistema.

**Tabla 29: Caso de prueba HU Autenticar usuario.**

Id del Escenario	Escenario	Resultado de la Prueba	Respuesta del sistema
EC1	Autenticación exitosa.	Satisfactoria	El sistema le concede el acceso al usuario
EC2	Autenticación fallida		El sistema no le concede el acceso al usuario. El sistema muestra un mensaje diciendo: "Credenciales invalidas, por favor inténtelo de nuevo"

**Tabla 30: Descripción de la HU Adicionar usuario.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Adicionar Usuario	Adicionar un usuario al sistema para que este puede acceder a él.

**Tabla 31 :Variables de la HU Adicionar usuario.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Usuario	Campo a llenar	No	Campo donde se introduce el nombre de usuario.
Contraseña	Campo a llenar	No	Campo donde se introduce la contraseña del usuario.

Rol	Campo a llenar	No	Campos donde se introduce el Rol del usuario adicionado.
Adicionar	Botón	No	Opción de Adicionar
Cancelar	Botón	No	Opción de Cancelar la Adición.

**Tabla 32: Caso de Prueba HU Adicionar usuario.**

Id del Escenario	Escenario	Usuario	Contraseña	Rol	Adicionar	Cancelar	Resultado de la Prueba	Respuesta del sistema
<b>EC1</b>	Selección a la opción Adicionar Usuario	V	V	V	N/A	N/A	Satisfactoria	Se muestra un formulario con los campos a llenar por el usuario.
<b>EC2</b>	Introduce los datos y selección	V	V	V	V	N/A		Regresa a la página principal mostrando

	a la opción Adicionar							el mensaje: Adición satisfactoria.
<b>EC3</b>	Campo incompleto	V	V	V	V	N/A		Se muestra un mensaje, "Rellene este campo".

**Tabla 33: Descripción de la HU Buscar comensal.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Buscar Comensal	Buscar un comensal en el sistema.

**Tabla 34: Variables de la HU Buscar comensal.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Buscar	Campo a llenar	SI	Campo donde se introduce el solapín del comensal que se desea buscar.
Buscar	Botón	No	Opción de Buscar

**Tabla 35: Caso de prueba HU Buscar comensal.**

Id del Escenario	Escenario	Solapín del comensal	Resultado de la Prueba	Respuesta del sistema
EC1	Introduce el número del solapín.	V	Satisfactorio	Se muestra los detalles del comensal.
EC3	Campo incompleto	V		Se muestra una tabla vacía donde deberían estar los datos de un comensal.

**Tabla 36: Descripción de la HU Adicionar comedor.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Adicionar comedor	Adicionar los datos de un comedor.

**Tabla 37: Variables de la HU Adicionar comedor.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Puerta	Campo a llenar	No	Campo donde se introduce el número de puerta del comedor.

Complejo	Campo a llenar	No	Campo donde se introduce el nombre del complejo comedor.
Adicionar	Botón	No	Opción de Adicionar
Cancelar	Botón	No	Opción de Cancelar la Adición.

**Tabla 38: Caso de prueba HU Adicionar comedor.**

Id del Escenario	Escenario	Puerta	Complejo	Adicionar	Cancelar	Resultado de la Prueba	Respuesta del sistema
<b>EC1</b>	Selecciona la opción Adicionar comedor.	V	V	N/A	N/A	Satisfactoria	Se muestra un formulario con los campos a llenar por el usuario.
<b>EC2</b>	Introduce los datos y selecciona la opción	V	V	V	N/A		Se muestra los detalles del nuevo comedor:

	Adicionar.						. –Puerta
<b>EC3</b>	Campo incompleto	V	N/A	V	N/A		Se muestra un mensaje, “Rellene este campo”.

**Tabla 39: Descripción de la HU Modificar comedor.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Modificar Comedor	Modificar todos o algún dato de un comedor y agregarlo en el sistema

**Tabla 40: Variables de la HU Modificar comedor.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Puerta	Campo a llenar	No	Campo donde se introduce el numero de la puerta del comedor.
Complejo	Campo a llenar	No	Campo donde se introduce nombre del complejo comedor.
Modificar	Botón	No	Opción de Modificar
Cancelar	Botón	No	Opción de Cancelar la Modificación



**Tabla 41: Caso de prueba HU Modificar comedor.**

Id del Escenario	Escenario	Puerta	Complejo	Modificar	Cancelar	Resultado de la Prueba	Respuesta del sistema
<b>EC1</b>	Selecciona la opción Modificar Comedor	V	V	N/A	N/A	Satisfactoria	Se muestra un formulario con los campos a llenar por el usuario.
<b>EC2</b>	Introduce los datos y selecciona la opción Modificar	V	V	V	N/A		Se muestra los detalles del nuevo Comedor: . –Puerta
<b>EC3</b>	Campo incompleto	V	N/A	V	N/A		Se muestra un mensaje, “Rellene este campo”.

**Tabla 42: Descripción de la HU Eliminar comedor.**

Nombre de Historia de Usuario	Descripción de la Funcionalidad
Eliminar comedor	Eliminar un comedor del sistema.

**Tabla 43: Variables de la HU Eliminar comedor.**

Nombre del campo	Clasificación	Valor nulo	Descripción
Eliminar	Botón	No	Opción de Eliminar
Cancelar	Botón	No	Opción de Cancelar la Eliminación.

**Tabla 44: Caso de prueba HU Eliminar comedor.**

Id del Escenario	Escenario	Eliminar	Cancelar	Resultado de la Prueba	Respuesta del sistema
<b>EC1</b>	Selecciona la opción Eliminar comedor	N/A	N/A	Satisfactoria	Se muestra un mensaje :Se eliminara este comedor.
<b>EC2</b>	Presiona el botón Eliminar	V	N/A		Se elimina el comedor del sistema.
<b>EC3</b>	Presiona el botón	V	N/A		Se cancela la operación.

	cancelar				
--	----------	--	--	--	--

1