Universidad de las Ciencias Informáticas

Facultad 6

Sistema de control de acceso automatizado para los laboratorios de la Universidad de las Ciencias Informáticas

Trabajo de diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor:

Aquiles Pérez Miranda

Tutores:

Msc. Iliannis Pupo Leyva

Msc. Denys Buedo Hidalgo

Ciudad de la Habana, Cuba, 2013.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

_____ _____

Aquiles Pérez Miranda Msc. Iliannis Pupo Leyva

_____

Msc. Denys Buedo Hidalgo

Resumen

En la Universidad de las Ciencias Informáticas (UCI) existe una gran afluencia de alumnos y profesores por lo que se hace difícil controlar la entrada y salida de las personas autorizadas, generando así un riesgo de seguridad en cuanto a información y tecnología. Debido a la problemática anterior es necesario el desarrollo de una herramienta que mantenga 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. En esta investigación se presenta el proceso de desarrollo de la primera versión del sistema de control de acceso llamado JKEEPER. 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, así como la generación de reportes a partir de la información obtenida de las funcionalidades descritas, para la toma de decisiones administrativas. Para llevar a cabo esta investigación se usó la metodología OpenUp y la integración del lenguaje Java con otros lenguajes como son PHP y JavaScript, utilizando una arquitectura n-capas. Este trabajo constituye un aporte al desarrollo de los sistemas de control de acceso en la UCI.

Palabras Claves: Control de acceso, sistemas de control de acceso, estaciones de trabajo, laboratorios, usuarios.

Índice

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) tienen, día a día, una mayor presencia en todos los aspectos de la vida laboral y personal, brindando un nuevo espacio de innovación en ámbitos como la industria, los servicios, la salud, la administración, el comercio y la educación. Los grandes avances de la tecnología informática, su efectividad y gran alcance social han creado una dependencia de su uso de manera global dando lugar a una fuente de conocimiento, entretenimiento y comunicación, insuperable por otro medio creado por la humanidad.

Las Tecnologías de la Información y la Comunicación, también conocidas como TIC, son el conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro. Abarcan un abanico de soluciones muy amplio. Incluyen las tecnologías para almacenar información y recuperarla después, enviar y recibir información de un sitio a otro, o procesar información para poder calcular resultados y elaborar informe. (1)

La nueva sociedad está en expansión y cada día se comparten más datos en la red, la tasa de innovación crece constantemente y con ella los cambios, amenazas y oportunidades que experimenta el espacio social en el que se desenvuelven los individuos. Los fallos informáticos han creado un nuevo elenco de problemas sociales, y así términos como crimen por ordenador, robo de software, piratas y virus informáticos, etc., son realidades cada día más corrientes y empiezan a representar un problema importante.

Debido a esto se hace necesario dotar a las empresas y organizaciones de mecanismos que permitan: gestionar usuarios y sus datos de identificación; asociar roles, perfiles y políticas de seguridad; y controlar el acceso a los recursos. Estos suelen estar integrados con mecanismos de autenticación que posibilitan el control del acceso lógico de los usuarios en los sistemas informáticos pues la ausencia de seguridad puede conllevar a pérdidas millonarias y riesgo de pérdida de información.

Las empresas sumidas en la problemática anterior, y en busca de una solución se estableció la ISO 27000 a nivel internacional para la adecuada gestión de la seguridad de la información, un sistema que aborda esta tarea de una forma metódica, documentada y basada en unos objetivos claros de seguridad y una evaluación de los riesgos a los que está sometida la información de la organización, además de que cada empresa ha creado asesores o directores de seguridad informática generando así un plan de seguridad informática, donde se refleja las estrategias de enfrentamiento a los posibles ataques en cuanto a seguridad de información se refiere.

Existen algunos países que se destacan en la producción y utilización de los controladores de acceso automatizados como son Estados Unidos, Inglaterra, Francia y China pues el control de acceso a instalaciones militares, centros de investigación,

universidades es de vital importancia. Los controladores de acceso más conocidos son los de las universidades norteamericanas llamado tecnología Kimaldi, los sistemas biométricos para instalaciones de gran importancia, la tecnología iClass y Javelin.

Cuba inmersa en informatizar a toda la sociedad ha puesto a disposición de todos, la enseñanza de la computación, el establecimiento de canales educativos y Universalización de la Educación Superior, ha creado los Institutos Politécnicos de Informática (IPI) y los Joven Club de Computación con el objetivo de llevar la informática a todo el pueblo. La reciente puesta en marcha del puerto del Mariel y su enorme cantidad de personal hizo posible el despliegue de un control de acceso mediante lectura de código de barra identificando al trabajador y su horario de trabajo.

La Universidad de las Ciencias Informáticas (UCI), surge en medio de este deseo de crecer en el desarrollo tecnológico, pues su objetivo principal es contribuir a la formación de profesionales que permitan la explotación de las nuevas tecnologías, donde se implementen ideas y proyectos que eleven el uso de esta rama a nivel empresarial y social. En la universidad existe una gran afluencia de estudiantes y profesores, es complicado mantener un control estricto de la entrada y salida del personal autorizado, lo que representa para la institución un riesgo de seguridad en cuanto a información y tecnología.

El área de los laboratorios en la UCI cuenta con aproximadamente 150 laboratorios, tiene como funciones primordiales brindar servicio, soporte e información a la docencia y la producción, los accesos a dichos laboratorios no son controlados, ni a las estaciones de trabajo directamente en cuanto a usuario y aplicaciones que este ha activado, un estudiante puede tener acceso a cualquier laboratorio y estación de trabajo, autenticándose con su cuenta de dominio UCI.

Para darle solución a la situación, se desarrolló en el año 2006 un sistema llamado Sistema de Control de Acceso a los Laboratorios de Producción (UCILAB), por la Dirección de los Laboratorios. Este software, fue creado con el objetivo de controlar el acceso a los laboratorios de la universidad. Se comenzó a poner en práctica en el año 2007, el controlador de acceso ofrece la funcionalidad de registrar la fecha y la hora en que los usuarios entraban y salían de los laboratorios.

UCILAB brinda además, la posibilidad de mostrar reportes de los accesos del personal a los laboratorios haciendo un análisis del personal que recibió ese servicio a través del control de acceso. A pesar del esfuerzo por mejorar la seguridad de los laboratorios de la UCI, aún existen brechas. Dicha aplicación permite el acceso del usuario al laboratorio pero no controla el acceso a las estaciones de trabajo directamente, ni las aplicaciones y procesos activados en cada una de ellas, esto da lugar a que no se tenga un estricto control de la adecuada explotación de los ordenadores.

El sistema además, no puede mapear las estaciones de trabajo en cuanto a dirección IP, los reportes no son confiables pues se desconoce las actividades del usuario dentro del laboratorio. Además, fue desarrollado en PHP y JavaScript, estos lenguajes están limitados a trabajar sobre el navegador siendo poco flexibles en cuanto a comunicación con el sistema operativo de la estación de trabajo y se desconocía las actividades del usuario en cada estación de trabajo. Debido a lo expuesto anteriormente el **problema a resolver** queda formulado de la siguiente manera: ¿Cómo controlar el acceso para los laboratorios de la UCI que permita la supervisión del uso de las tecnologías?

Por tanto el **objeto de estudio** de este trabajo lo constituyen los sistemas automatizados de control de acceso y horarios. El **campo de acción** queda enmarcado específicamente en los sistemas automatizados de control de acceso y horarios para laboratorios computacionales en la UCI. La **idea a defender** que se establece como base de la investigación queda formulada de la siguiente manera: La creación de un sistema de control de acceso automatizado para los laboratorios de la UCI permite supervisar el uso de las tecnologías.

El **objetivo general** es desarrollar un sistema de acceso automatizado para los laboratorios de la UCI que permita controlar el uso de las tecnologías

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

- Realizar el estudio del arte relacionado con el control automatizado de acceso y horarios.
- Analizar y diseñar un sistema de control de acceso automatizado para los laboratorios de la UCI.
- Implementar un sistema de control de acceso automatizado para los laboratorios de la UCI.
- Validar la propuesta de solución a través de pruebas funcionales y de aceptación.

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

- Realización de un estudio de la bibliografía necesaria para conocer el estado actual de los sistemas de control automatizado de acceso y horarios.
- Selección de la metodología, las herramientas y las tecnologías a utilizar para el desarrollo de la solución propuesta.
- Realización del análisis y el diseño del sistema de control de acceso automatizado para los laboratorios de la UCI.
- Implementación del agente que controla el acceso a las estaciones de trabajo e intercambia información con el controlador de entrada al laboratorio.

- Implementación de la aplicación web que funciona como controlador de entrada y gestiona el acceso a los laboratorios a través del solapín institucional.
- Validación de la solución propuesta mediante pruebas funcionales y de aceptación.

Diseño Metodológico

Los métodos teóricos de la investigación científica usados para cumplir las tareas son:

- **Método Lógico:** Se usó para estudiar de forma analítica la trayectoria histórica de los fenómenos, evolución y desarrollo de los procesos de control de acceso.
- **Modelación:** Se modelaron diagramas para facilitar la implementación del sistema, como parte de la solución propuesta.
- **Analítico-Sintético:** Utilizado para analizar la bibliografía y realizar una síntesis de la misma.
- **Inductivo-Deductivo**: Utilizado para analizar la bibliografía y realizar una síntesis de la misma.

Estructura del documento

Este documento ha sido conformado por: Resumen, Introducción, 4 capítulos que conforman la integridad de la tesis, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Anexos.

Capítulo 1 Fundamentos Teóricos: Describe cómo se realiza el proceso de control de acceso a los laboratorios en la Universidad actualmente. En él se mencionan los principales problemas que generaron la necesidad del cambio; se obtienen los objetivos generales y específicos a cumplir. Trata la situación de las tecnologías a utilizar en el desarrollo de la aplicación, y se explican los conceptos principales que se van a tratar.

Capítulo 2 Características de la Aplicación: Aborda las características de la aplicación a desarrollar. Se define el modelo de negocio, los requisitos funcionales, y no funcionales que requiere, los actores del sistema, diagramas de caso de uso del sistema y el patrón de casos de uso utilizado.

Capítulo 3 Análisis y Diseño de la Aplicación: Se describen los patrones de diseño empleados. Se presentan los diagramas de clases del diseño, diagramas de interacción y diagrama de despliegue de la aplicación con el objetivo de tener una idea de cómo quedará.

Capítulo 4 Implementación y Análisis de los resultados de la aplicación: Presenta el diagrama de componentes, se muestran los resultados obtenidos y los casos de prueba para validar la hipótesis.

Capítulo 1 Fundamentos Teóricos

Introducción

Un sistema de control de acceso protege la confidencialidad, integridad y disponibilidad de los recursos mediante mecanismos que dificultan la entrada de usuarios no autorizados a los mismos. El control de acceso generalmente consta de dos pasos:

- En primer lugar, la autenticación, que identifica al usuario o la máquina que trata de acceder a los recursos.
- En segundo lugar, ceder los derechos, es decir, la autorización, que dota al usuario de privilegios para poder efectuar ciertas operaciones con los datos protegidos, tales como leerlos o modificarlos.

En este capítulo se hace un estudio de estado del arte sobre los antecedentes de los controladores de acceso, así como algunos de los principales sistemas a nivel internacional y nacional. Se describen las principales características de las herramientas y frameworks[1]a utilizar y se selecciona la metodología a usar en la solución propuesta.

1.1 Seguridad de los sistemas

La necesidad de mantener la integridad de la información y de proteger los activos de las TICs, requiere de un proceso de administración de la seguridad. Este proceso incluye el establecimiento y mantenimiento de roles y responsabilidades de seguridad, políticas, estándares y procedimientos. La administración de seguridad también incluye realizar monitoreos de seguridad y pruebas periódicas así como realizar acciones correctivas sobre las debilidades o incidentes de seguridad identificados.

Una efectiva administración de la seguridad protege todos los activos de las TICs para minimizar el impacto causado por vulnerabilidades o incidentes de seguridad. El aspecto de la seguridad es un elemento muy importante en la gestión de los procesos de negocio de una organización. La seguridad de la información persigue proteger los recursos de posibles accesos y modificaciones no autorizadas (2)

Los principales objetivos de la seguridad de la información se pueden resumir en:

- ***Confidencialidad:*** Describe el estado en el cual la información está protegida de revelaciones no autorizadas.
- ***Integridad:*** Significa que la información no ha sido alterada o destruida por una acción accidental o por un intento malicioso. (3)

[1] **Frameworks:** Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

- ***Disponibilidad:*** Referencia al hecho de que una persona autorizada pueda acceder a la información en un apropiado período de tiempo. Las razones de la pérdida de disponibilidad pueden ser ataques o inestabilidades del sistema.
- ***Responsabilidad:*** Asegurar que las acciones realizadas en el sistema por un usuario se puedan asociar únicamente a ese usuario, que será responsable de sus acciones. Es decir, que un usuario no pueda negar su implicación en una acción que realizó en el sistema. (4)

1.2 Autenticación.

Por autenticación se entiende cualquier método que permite comprobar de manera segura cierta característica de un objeto, como su origen, su no manipulación, su identidad (5). Existen dos grandes grupos dentro de los métodos de autenticación:

- ***Autenticación de mensaje:*** Garantiza la procedencia de un mensaje, identificando al usuario que lo emitió. Por definición, este tipo de autenticación involucra también la integridad de los datos, es decir, permite detectar si el contenido del mensaje fue accidental o maliciosamente alterado.
- ***Autenticación de usuario:*** Garantiza que el usuario es quien afirma ser. En este proceso se corrobora la identidad del usuario en tiempo real, es decir, en el mismo momento que se están comunicando el verificador y el usuario.

1.2.1 Autenticación de mensaje.

Las técnicas más utilizadas para proveer autenticación de mensaje son:

- ***MAC (Message Authentication Code):*** Los códigos de autenticación de mensajes fueron diseñados especialmente para aplicaciones donde se requiere la integridad de los datos (pero no necesariamente su privacidad). Un MAC es un tipo de función unidireccional que toma dos entradas (un mensaje y una clave secreta) y retorna una cadena de tamaño fijo, siendo imposible volver a obtener la misma salida sin conocer la clave. Así, el emisor de un mensaje M, usa la clave secreta compartida K para calcular un MAC *hK (M)* sobre el mensaje y envía al receptor M y *hK (M).*

  El receptor determina la identidad de la fuente del mensaje (utilizando, por ejemplo, un campo identificador en el texto no cifrado) y separa el MAC de los datos. Luego, calcula un MAC sobre estos datos, utilizando una clave MAC compartida, y compara el MAC calculado con el recibido. Si estos valores coinciden, el receptor interpreta que los datos son auténticos y que no fueron modificados durante su transmisión.

- ***Firma Digital:*** Simula una firma real. Para ello, se genera una prueba digital que solo el emisor de un mensaje puede crear, pero que todos pueden identificar como perteneciente a ese emisor. Un cifrado utilizando la clave privada del emisor sirve como firma, porque sólo el propietario de la clave privada puede crearlo y todos pueden verificarlo con la clave pública correspondiente. El cifrado (firma) puede aplicarse a todo el mensaje o a un pequeño bloque de datos que sea función del mensaje, por ejemplo, el valor hash.

  Por tanto, el emisor A envía al receptor B el mensaje M y la firma calculada sobre el hash del mensaje ESKA (H (M)), donde SKA es la clave privada del emisor, E es el algoritmo de cifrado asimétrico utilizado y H (M) es el hash del mensaje. El receptor debe entonces obtener la clave pública del emisor PKA y realizar una operación de cifrado sobre la firma del mensaje, con lo que tendría H´ (M). Posteriormente, B debe calcular el hash del mensaje y compararlo con H´ (M). Si los dos resultados coinciden, se corrobora la autenticidad y la integridad del mensaje. (6)

1.2.2 Autenticación de usuario.

Las técnicas de autenticación de usuario pueden clasificarse en tres categorías, dependiendo de qué deba presentar el usuario para demostrar su identidad:

- ***Algo conocido:*** El usuario debe demostrar que conoce algún tipo de información secreta, por ejemplo, una palabra clave, un número de identificación personal (PIN), una clave privada.

- ***Algo que Posee:*** El usuario requiere para identificarse algún tipo de dispositivo, como: una tarjeta magnética, una tarjeta inteligente, un generador de contraseñas o una llave electrónica.
- ***Algo Inherente (Propio del individuo):*** Aquí se requiere alguna característica fisiológica del usuario (biometría), por ejemplo: huellas dactilares, la voz, la retina, el iris. Esta categoría puede considerarse como un caso particular de la anterior, donde el dispositivo es el propio usuario. (6)

Las técnicas de autenticación de usuario basadas en algo conocido, se pueden clasificar a su vez en tres categorías de acuerdo al nivel de seguridad que ofrecen:

- ***Autenticación Débil:*** Es el tipo de autenticación más extendido, y al que los usuarios ya están acostumbrados. Aquí, una contraseña o palabra clave (*password*), asociada a cada usuario, es el secreto compartido entre el usuario y el sistema. Para identificarse, el usuario introduce un identificador (*login*) y su contraseña. El sistema comprueba entonces que esa contraseña coincida con la almacenada para dicho identificador. Ya que las contraseñas son fijas y

reutilizables, si un atacante se entera de la clave de un usuario, puede suplantar su identidad fácilmente.

Para protegerse de los posibles ataques por fuerza bruta, de los que pueden ser víctima estos sistemas, se limita el número de intentos que el usuario puede hacer desde un terminal concreto y se introducen retardos cuando la contraseña introducida es errónea.

- ***Autenticación Fuerte:*** Para este tipo de autenticación una entidad (el demandante) prueba su identidad ante otra entidad (el verificador) demostrando que conoce un secreto asociado con su identidad, pero sin revelar dicho secreto al verificador. Esto se hace respondiendo a un desafío variable. Dicha respuesta es típicamente un número elegido por el verificador (aleatoria y secretamente) al inicio del protocolo. Si un atacante obtiene la respuesta al desafío de un usuario determinado, esa información no le será útil para suplantar posteriormente a ese usuario, ya que el desafío será diferente.

Los mecanismos de autenticación fuerte basados en criptografía simétrica, requieren que el demandante y el verificador compartan una clave secreta. Cuando el sistema cuenta con un gran número de usuarios se suele usar un servidor confiable en línea, con el que cada usuario comparte una clave. Este servidor actúa a manera de Hub[2], proporcionando una clave de sesión común a las dos partes comunicantes cada vez que quieran autenticarse una con la otra. El protocolo Kerberos provee autenticación fuerte basada en cifrado simétrico e involucra el uso de una tercera parte de confianza en línea.

En los mecanismos de autenticación fuerte basados en criptografía asimétrica, el demandante demuestra que posee su clave privada, bien sea descifrando un desafío cifrado con su clave pública o firmando digitalmente el desafío. Para no comprometer la seguridad del sistema, el par de claves usado por estos mecanismos no debe ser usado con otros propósitos. La Infraestructura de Clave Pública (PKI) utiliza autenticación fuerte basada en criptografía asimétrica para proveer sus servicios de seguridad.

- ***Autenticación de Conocimiento Nulo:*** Los protocolos de conocimientos nulos (**ZK: *Zero- knowledge***) le permiten al probador (el demandante) demostrar el conocimiento de un secreto sin revelar ninguna información útil para el verificador (más allá de la que el verificador es capaz de deducir antes de ejecutar el protocolo). Los sistemas de prueba interactivos son un ejemplo de este tipo de protocolos.

  Aquí el objetivo del probador es convencer al verificador de que posee algún secreto, contestando correctamente preguntas que requieren del conocimiento de

[2]**Hub:** Término en inglés con el que se denomina al concentrador, es un dispositivo que se utiliza como punto de conexión entre los componentes de una red de área local.

ese secreto para ser respondidas. Estos protocolos emplean técnicas asimétricas pero no utilizan firmas digitales o cifradas con clave pública. Tampoco usan cifradores de bloques, números de secuencia ni sellos de tiempo (*timestamps*). (6)

1.2.3 Arquitecturas de Autenticación

Las técnicas de autenticación descritas anteriormente implican algún tipo de clave o secreto compartido entre las partes comunicantes. Debe existir, por tanto, una infraestructura de seguridad que se encargue de la generación, distribución y gestión de los secretos. Esta infraestructura de seguridad brindará una base segura a toda la organización y deberá ser accesible por todas las aplicaciones y objetos de la organización que necesiten seguridad. (7)

Durante un proceso típico de autenticación, el usuario presenta su credencial (por ejemplo: *login*y *password*) o el resultado de una operación criptográfica que involucre su credencial, a la autoridad de autenticación. Ésta valida la credencial usando los datos guardados en la base de datos. Si la credencial proporcionada por el usuario y la guardada en la base de datos coincide, o si el resultado de la operación criptográfica sobre la credencial guardada en la base de datos es igual a la información suministrada por el usuario, la identidad del usuario es considerada auténtica. (8)

Sin embargo, el usuario debe compartir una credencial con cada dominio y autenticarse cada vez que quiera acceder a algún recurso. Dada la diversidad de plataformas de computación, sistemas operativos y de software de control de acceso, lo más deseable es registrarse en múltiples sistemas una vez y simultáneamente a través de una sola transacción. Nace aquí el registro único (SSO – *Single Sign - On*). Con SSO el usuario debe autenticarse solo una vez para acceder a múltiples sistemas.

El registro único puede usarse en infraestructuras con diversas autoridades de autenticación, es decir, implementadas en diferentes plataformas y gobernadas por diversas organizaciones. Las arquitecturas más simples son las que usan un conjunto de credenciales. Las dos más importantes son:

- ***Sistemas Basados en Testigos (Token - Based):*** En una arquitectura basada en testigos, los usuarios obtienen un testigo temporal después de autenticarse exitosamente ante su *Trusted Third Party* (TTP). Este testigo puede ser guardado en el dispositivo del usuario y reutilizado para probar su identidad ante las TTP de dominios de autenticación secundarios. Para validar el testigo de un usuario, estas TTP usan métodos criptográficos basados en claves secretas establecidas previamente entre ellas y la TTP del dominio de autenticación primario. Estas claves criptográficas permiten establecer una relación de confianza entre diferentes dominios de autenticación.

- ***Sistemas basados en PKI (PKI - Based):*** En esta arquitectura, los usuarios se registran primero ante una autoridad de autenticación confiable, la Autoridad Certificadora (AC). Durante el proceso de registro los usuarios se identifican a través de un conjunto de credenciales. El software del usuario genera un par asimétrico de clave, y la clave pública es ofrecida a la autoridad certificadora para su certificación. Después de recibir las credenciales del usuario y la clave pública, la autoridad certificadora verifica si las credenciales son válidas. Si es así, genera un certificado de clave pública y se lo retorna al usuario.

  Este certificado y la clave son guardados de manera segura en el dispositivo del usuario u otro medio, como tarjeta inteligente. Estos son usados para probar la identidad del usuario ante otras autoridades de certificación en solicitudes de autenticación posteriores. En esta arquitectura, la relación de confianza entre la autoridad de certificación primaria y las secundarias se establece a través de un certificado expedido por la AC primaria y la AC secundaria. (8)

1.3 Autorización.

La autorización está estrechamente ligada con la autenticación. Una vez que el usuario ha validado su identidad para acceder a algún recurso, es necesario restringir sus acciones de acuerdo a quién es, que está tratando de hacer (9). La autorización dota al usuario de privilegios para poder efectuar ciertas operaciones con los datos o recursos protegidos. La autorización está compuesta por varios elementos como son:

- Usuarios
- Privilegios
- Recursos
- Reglas que definen como los usuarios adquieren o pierden privilegios sobre determinados recursos.

Los sistemas más comunes que usan la autorización son los sistemas operativos, las comunidades de red, corporaciones o sistemas empresariales de gran tamaño (10).

1.4 Sistemas de control de acceso automatizados

Desde los orígenes de la humanidad se ha tratado de controlar el acceso de las personas a lugares restringidos. Un control de accesos es un dispositivo que tiene por objeto impedir el libre acceso del público en general a diversas áreas que denominaremos protegidas. (11) El acelerado desarrollo de la informática y las tecnologías trajo consigo una notable evolución en los sistemas del control de acceso. En un mundo desarrollado, es de suma importancia proteger el acceso a los recursos y evitar una mala manipulación de los mismos, que podría causar pérdidas considerables. Surgiendo problemas, los cuales llevan consigo dos conceptos que muy a menudo se mezclan de manera difusa: la

autentificación (pretende establecer quién eres) y la autorización (establece que podemos hacer).
Estos conceptos parecen ir ligados de forma indisoluble pero no siempre ocurre de esta manera, por lo que hay que tener claro estos conceptos antes mencionados.

Para impedir pérdida de recursos, los sistemas de control tienen soluciones las cuales llevan consigo la utilización de los tipos básicos de control de acceso, que tienen filosofías diametralmente opuestas. El objetivo principal de un Sistema de Control de Accesos e Identificación de Personal es el de controlar y monitorear el flujo del personal en un inmueble, decidiendo quien entra o sale, a donde y a qué horas lo puede hacer (12).

En términos técnicos o lógicos el acceso es la interacción entre un sujeto y un objeto que resulta en un flujo de información de uno al otro. El sujeto es la entidad que recibe o modifica la información o los datos contenidos en los objetos, puede ser un usuario, programa, proceso, etc. Un objeto es la entidad que provee o contiene la información o los datos, puede ser un fichero, una base de datos, una computadora, un programa, una impresora o un dispositivo de almacenamiento.

Incluye autenticar la identidad de los usuarios o grupos y autorizar el acceso a datos o recursos. Los controles de accesos son necesarios para proteger la confidencialidad, integridad y disponibilidad de los objetos, y por extensión de la información que contienen, pues permiten que los usuarios autorizados accedan solo a los recursos que ellos quieren para realizar sus tareas (13).

Un factor fundamental para determinar el funcionamiento de todo entorno de autorización es la definición del modelo de control de acceso que se va a establecer. Los mecanismos utilizados para restringir el acceso a los recursos son generalmente de una (o algunas veces la combinación) de dos formas (14).

- ***Control de Acceso Discrecional (DAC):*** Le deja las decisiones de control de acceso al propietario del recurso, de manera que es este quien decide que sujetos pueden realizar determinadas acciones sobre los recursos poseídos. Así, un sujeto con permisos de acceso puede otorgarlos a otro sujeto.
- ***Control de Acceso Obligatorio (MAC):*** Este control de acceso se basa en reglas establecidas por una autoridad central. MAC restringe el acceso a los objetos basándose en la sensibilidad de la información que estos contienen (representada por una etiqueta) y en la autorización formal de los sujetos para acceder a dicha información. Los objetos son considerados como entidades pasivas que almacenan información, mientras que los sujetos son entidades activas que realizan peticiones de acceso a los objetos.

Existen diversos modelos de control de acceso de tipo DAC y/o MAC (15). Los más comunes son:

- ***Control de Acceso Basado en Identidad (IBAC):*** En este caso los permisos de acceso se asocian directamente al identificador del sujeto, es decir, el nombre del usuario. Por tanto se garantiza el acceso al recurso solo cuando exista dicha asociación. Con IBAC no se asocian etiquetas de seguridad a los usuarios, por lo que este es principalmente un mecanismo de control de acceso discrecional.

  Un ejemplo de IBAC son las Listas de Control de Acceso (ACLs), encontradas comúnmente en sistemas operativos y servicios de seguridad en red. Una Lista de Control de Acceso contiene los identificadores de los usuarios junto con sus derechos de acceso a un recurso determinado, como leer, escribir, ejecutar. Esta estructura básica de autorización únicamente extiende el concepto de autenticación, ya que si el usuario no puede autenticarse correctamente ante el guardián del recurso, su solicitud de acceso es denegada.

  Entre más usuarios soliciten el acceso a un recurso más identificadores contendrá la ACL, lo que dificulta el manejo de estas listas y las hace una alternativa poco escalable. Por otra parte, la decisión de control de acceso no depende de alguna función o característica de la organización a la que pertenece el usuario sino solamente de los identificadores, así que su uso es inapropiado a nivel empresarial.

- ***Control de Acceso Basado en Roles (RBAC):*** Restringe el acceso a los recursos basándose en la función o rol que desempeña el sujeto dentro de la organización. Los permisos para acceder a un recurso son asignados a cada rol, en lugar de asociarlos directamente al identificador del sujeto. El RBAC es escalable y reduce significativamente la cantidad de información de administración necesaria, ya que los permisos no son asignados constante e individualmente a cada usuario. RBAC es primordialmente un mecanismo de control de acceso discrecional. Generalmente no tiene en cuenta las características de los recursos (más que sus identificadores) y no obtienen ninguna información relevante de seguridad del entorno.

- ***Control de Acceso Basado en Atributos (ABAC):*** En ABAC los privilegios son establecidos en base a la colección de atributos que posee el usuario y una política que los determina. ABAC es la convergencia natural de los modelos de control de acceso IBAC y RBAC. La representación de las políticas en ABAC es semánticamente más rica y expresiva. Además posee una mayor granularidad ya que puede basarse en cualquier combinación de atributos de sujeto, de recursos y de entorno.

- ***Control de Acceso Basado en Mallas (LBAC):*** En este caso una colección totalmente ordenada de etiquetas de seguridad se combina con un grupo de categorías y forman una malla. Con ellos se obtienen un conjunto de clases de seguridad que varía desde la más baja a la más alta (16). Generalmente el modelo LBAC es manejable cuando existe un número relativamente pequeño de etiquetas de seguridad y categorías, por lo que es solo efectivo para ciertos escenarios de seguridad poco granulados y carentes de flexibilidad y escalabilidad. LBAC es un mecanismo de control de acceso obligatorio.

1.5 Tendencias actuales de los sistemas de control de acceso automatizados

Existen innumerables sistemas de control de acceso automatizados que le dan seguridad y protección a diversas instalaciones, desafortunadamente se vive en tiempos donde los riesgos incrementan, por lo que surge la necesidad de utilizar estos sistemas. Son numerosos los sectores y las aplicaciones en los que se tiene en cuenta el control de acceso, tanto en entornos domésticos como casas, edificios, control de acceso comercial e industrial, control de acceso para hoteles, balnearios, centros sanitarios, universidades, centros deportivos, puertos y otras áreas reservadas o de seguridad.

1.5.1 Hardware utilizado en los sistemas de control de acceso

- Claves por teclado

Es una de las vías más económicas pero la más insegura, hace mucho tiempo las instituciones han dejado de usarlas, hasta el momento no se han generado aplicaciones o investigaciones que puedan sugerirla como una vía segura.

- Tarjetas de Banda Magnética

Es una de las tecnologías más difundidas, se utiliza en los sistemas de tarjetas de crédito y compra. La ventaja es su difusión, popularidad y el bajo costo. Es uno de los medios de identificación más vulnerables, pues la banda magnética de la tarjeta debe ser tratada con cuidado para evitar que se raye, por ello no son recomendables en ambientes industriales.

- Tarjetas de código de barras

El código de barras es una tecnología de identificación automática. Permite recolectar datos con precisión y rapidez. Un código de barras consiste en una serie de barras adyacentes paralelas y espacios. Los diseños predeterminados de anchura se utilizan para codificar datos en el código. Para leer información en un símbolo de código de barras, un dispositivo de lectura, tal como un lápiz óptico, se desliza a través del símbolo de un lado al otro, la anchura de barras y los espacios son analizados por el decodificador del lector, y los datos originales se recuperan. En la UCI se hace uso de las tarjetas de código de barras en forma de solapín institucional por su fácil funcionamiento, lo cual identificará al trabajador desde cualquier lugar en la zona universitaria además es el método de control de acceso a usar en la solución propuesta de esta investigación.

- Touch Memories

Comúnmente se los denomina llave electrónica y brindan un alto nivel de seguridad, ya que son altamente resistentes al desgaste, siendo ideales para ambientes industriales en donde la probabilidad de falla, vandalismo o sabotaje sea alta, aunque no son recomendables para ambientes con alto grado de generación de corriente estática (Ej.: oficinas con mucha alfombra y ambientes muy secos). Su tecnología de avanzada evita la posibilidad de duplicarlas. En precio hay que tener en cuenta que son unos de los medios más caros, sin embargo, nunca se desgastan, como puede suceder con una tarjeta magnética, ya que en lo que al lector respecta, es también de acero.

Los dispositivos de lectura denominados "*Touch Memories*" son memorias de contacto inalterables con la apariencia externa de una pila de calculadora que contienen en su interior un código irrepetible y que sirve para identificar a su portador. Se provee con un llavero para su fácil operación. (17)

- Tarjetas de proximidad

Las tarjetas de proximidad son dispositivos que están contantemente enviando señales al lector para saber la posición exacta de cada una en todo momento. También se denominan dispositivos RFID (*Radio Frequency Identification*) cuyo propósito es el de transmitir la identidad de un objeto mediante ondas de radio. Una de las ventajas del uso de radiofrecuencia (en lugar, por ejemplo, de infrarrojos) es que no se requiere visión directa entre emisor y receptor (18).

- Control de Acceso a Universidades.

Kimaldi[3], ofrece un sistema de control de acceso a Universidades. El sistema incorpora un potente programa que, abarca todo el control específico de todos los tipos de registros para las entradas y/o salidas, gestiona el acceso por régimen jerárquico, por franjas horarias dentro de un calendario anual, el registro del histórico de los accesos no autorizados y los intentos de uso fraudulento. La actualización dela base de datos no exige conocimientos técnicos específicos, por tanto esta actividad resulta ágil y de fácil manejo. Aplicación múltiple de control de accesos y alarmas mediante radiofrecuencia.

El sistema de seguridad y control visual son proporcionados por los mismos terminales de acceso que son capaces de activar puntualmente las cámaras. El control de alarmas de las puertas de salida de emergencia del edificio principal, permiten visualizar por pantalla desde el centro de gestión cualquier incidencia que se produzca en cualquier salida de emergencia, así como el estado en que puede quedar (abierta o cerrada), y todos los registros de incidencias. (19) Los productos que se utilizan para esto son:

- Tarjeta Kimaldi Ndcan Max
- Lector proximidad Kimaldi RD125K
- Concentrador Kimaldi KMD905 v2

❖ Sistemas biométricos

Su funcionamiento se basa en la lectura o reconocimiento de alguna parte del cuerpo humano; de la huella dactilar, geometría de la mano, frecuencia de la voz, por la retina o reconocimiento facial; eliminando por completo el uso de las tarjetas. Los más conocidos son los lectores de huellas digitales, geometría de la mano e iris del ojo.

Sus desventajas son la velocidad y el precio, además, deben ir acompañados de un teclado (para anteponer un código para acelerar el proceso de búsqueda), y por último la poca posibilidad de ser autónomos (generalmente por su complicada lógica se ven obligados a trabajar con un software de análisis y una PC conectada directa al lector, lo cual es poco práctico y más caro), pero seguramente con el tiempo se irán superando estas dificultades y en un futuro no lejano, llegarán a ser una opción más asequible en el mercado.

❖ Tecnología iCLASS

La tecnología iCLASS garantiza una elevada seguridad con autenticación mutua entre tarjeta y lector, transferencia de datos codificados y llaves diversificadas de 64 bits para capacidades de lectura/escritura. La tecnología lectura/escritura de la plataforma abre el

[3] **Kimaldi Electronics:** es una empresa dedicada a la fabricación, diseño y comercialización de productos dentro del mundo de la Identificación de personas.

camino para nuevas funcionalidades, a la vez que soporta sistemas anteriores para una fácil migración.

La tecnología inteligente iCLASS de HID ofrece una alta frecuencia de 13,56 MHz que proporciona una interoperabilidad versátil en aplicaciones como el control de acceso, la seguridad de acceso a la red, pago sin efectivo, el control de horarios, la gestión de eventos y la identificación biométrica (20)

- Javelin de AMAG Technology

El Javelin es uno de los primeros lectores de tarjetas en el mercado en ofrecer una pantalla gráfica de cuatro líneas con la opción de personalizar texto, disponible en 10 idiomas. Una barra de luz brillante en el LED tricolor muestra claramente el estado de la puerta. Para quienes requieran una flexibilidad completa, la línea Javelin también ofrece un modo gráfico que proporciona el estado del lector de tarjetas y las instrucciones del usuario a través de imágenes claras.

Otro beneficio del lector de tarjetas son las grandes teclas táctiles bien espaciadas con la habitual tecla guía en la quinta tecla como una ayuda táctil. Además de las ayudas visuales, el lector Javelin también proporciona anuncio sonoro, lo que brinda una secuencia ascendente y descendente de la función de los sonidos cuando se presente una tarjeta de manera exitosa. (21)

1.5.2 Software de sistemas de control de acceso

A continuación se nombrarán algunos de estos sistemas, los que han sido desarrollados e implantados en otros países, así como en la propia Universidad.

- Sistema de control de acceso “*Easy Way”*

El sistema de control de Accesos *Easy Way* es un seguro método destinado a controlar el ingreso y egreso de personas a todas las áreas de la empresa (es decir, control de personal). 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, 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, lectores de proximidad HID. (22)

- 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. (23)

- Sistema de control de acceso "*Altus Accesos*"

Altus Accesos es un sistema creado para gestionar e integrar informáticamente las necesidades de una empresa relacionadas con el control de accesos de sus empleados o de personas ajenas en sus edificios y delegaciones. El sistema posee las siguientes funcionalidades:

- **Gestión de usuarios y perfiles:** Se permite crear diferentes usuarios y perfiles para restringir el acceso a las diferentes partes de la aplicación así como a los diferentes informes. No existe limitación ni en el número de usuarios ni de perfiles a crear.
- **Gestión de empleados, datos personales, departamentos:** Se podrá detallar la información de cada empleado para un completo control y archivo de cada uno. No existe limitación en el número de empleados.
- **Control de Visitas:** Control de visitas mediante asignación de tarjetas a nuevas visitas, control de devolución de tarjetas, zonas de acceso, toma de datos personales, seguimiento de visitas.
- **Control de Presencia:** Módulo que permite la integración con el control de accesos y comparte la base de datos de personal para realizar un control de presencia, seguimiento de fichajes, listados varios de puntualidad, faltas de asistencia, horas extra, mantenimiento y asignación de calendarios laborales, etc.
- **Informes:** El sistema genera informes variados tanto de control de accesos, asistencia, control horario (presencia), incidencias, listado de personal, etc. (24)

Dentro de los sistemas automatizados relacionados con el tema de estudio que se presentan en la UCI, los más conocidos e importantes son:

- Sistema de Identificación.

Este sistema brinda un servicio de certificación de identidad a otros sistemas informáticos, como los que son para el control del acceso. Tiene almacenados los datos de todo el personal que labora y estudia en la Universidad: 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 frameworks tales como Spring Framework .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.

- Sistema de Control de Acceso a los Comedores.

Mediante este sistema se controla en los comedores de los diferentes Complejos Alimenticios 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 delos comedores y puertas a los mismos, además de ofrecer reportes como cantidad de comensales que han pasado y desglosarlo por puerta o por tipo. Esta aplicación, a pesar de incorporar tecnologías para el acceso a datos como un novedoso motor de base de datos orientado a objetos "*Database 4(for) objects"* (DB4O), tecnología a utilizar en el sistema que se desea desarrollar, no es de gran de utilidad debido a que la misma maneja información enfocada a la gestión del proceso logístico alimenticio, información que no es relevante para el sistema a desarrollar.

- 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 universidad, cada una de ellas específica para el área productiva donde se encuentra, lo que hace que no exista una

base de datos centralizadas con todos los datos referentes a todos los laboratorios de producción, a pesar de que son aplicaciones que no están bien concebidas ni con una amplia documentación. Sin embargo, la aplicación a desarrollarse es capaz de gestionar toda la información que manejan dichas aplicaciones de forma centralizada.

- 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. El sistema fue desarrollado en C# preferentemente para sistemas operativos propietarios.

- 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.

El controlador de acceso se implementó sobre PHP y consume los servicios UCI como son LDAP, WebServices de identificación, además de comunicarse con un firewall a nivel de red llamado PFSense, el objetivo principal de la aplicación es el control físico de las personas que acceden a los laboratorios.

1.6 Valoración de los sistemas investigados

Después de haber hecho un estudio de los sistemas de control de acceso comentados, se han logrado identificar algunas de las funcionalidades y características más comunes de estos tipos de software, como gestión de puertas, control de horarios y permisos, manejo de registro, recuperación de fichajes, identificación y clasificación del personal, creación de credenciales.

En el caso de los controladores de acceso internacionales como lo son *“Easy Way”*, *“Altus Accesos” y “Arquero”,* son sistemas propietarios con un alto costo de adquisición además son aplicaciones personalizadas a clientes específicos en cada empresa, en cuanto a los sistemas nacionales descritos anteriormente todos permiten el acceso del personal al local pero ninguno controla directamente las actividades del usuario dentro de la

institución. Debido a esto se asume en la investigación propuesta la tecnología de código de barras ya existente en la UCI para el proceso de autenticación y autorización.

1.7 Herramientas y tecnologías para el desarrollo de la solución propuesta

- Herramienta CASE: Rational Rose

Rational Rose Enterprisees una de las poderosas herramientas bajo licencia propietaria para el modelado UML producidas por IBM. Soporta además, la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++® y Visual Basic®. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

- Características adicionales incluidas:
- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++.
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos
- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables
- Soporte Enterprise Java Beans™ 2.0
- Capacidad de análisis de calidad de código
- El *Add-In* para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos
- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-*compliant*.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo (25).

- Herramienta CASE: Visual Paradigm

La tecnología CASE (*Computer Aided Software Engineering*, Ingeniería de software asistida por computadoras) supone la automatización del desarrollo de software,

contribuyendo a mejorar la productividad en el desarrollo de sistemas de información. Automatiza además la documentación, la generación de código, el chequeo de errores y la gestión del proyecto permitiendo así la reutilización y la portabilidad del software y la estandarización de la documentación.

Visual Paradigm es una herramienta CASE profesional libre que soporta todo el ciclo de vida del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones.

Entre sus características principales esta que permite el control de versiones, consta de un entorno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso. También permite el despliegue de EJB (Enterprise java Beans), la construcción de diagramas de flujo de datos, la generación de objetos Java desde la base de datos, la transformación de diagramas de Entidad-Relación en tablas de base de datos. Posee un generador de informes, la reorganización de las figuras y conectores de los diagramas y un editor de figuras (26).

En la solución propuesta se usará esta herramienta por las ventajas que brinda como son el diseño previo de clases y modelos, integración con el control de versiones, completamiento de código con respecto a frameworks usados y el diseño de la base de datos además de la integración con el lenguaje seleccionado para su implementación.

- Eclipse

Gran parte de la programación de Eclipse fue realizada por IBM (International Business Machines) antes de que se creara el proyecto Eclipse como tal. Con la aparición de Java en la década de los 90, su rápida expansión y sus ventajas con miras a un internet en plena expansión obligaron a IBM a plantearse la construcción de una plataforma basada en Java. Surgió así el proyecto Eclipse. Al inicio, Eclipse no era muy conocido en la comunidad internacional, por lo que el consorcio de la IBM se mostró renuente a invertir en la plataforma; por lo que a finales de 2001 IBM adoptó una licencia de código abierto para incrementar el desarrollo del mismo y acelerar la acogida por la comunidad de desarrolladores.

Para ese entonces IBM puso el proyecto Eclipse en manos de un consorcio (Eclipse.org) de empresas fabricantes de herramientas de software. Originalmente la junta directiva del consorcio incluía a Borland, MERANT, IBM, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft y Webgain. Con el tiempo el número de miembros fue aumentando y ya en el 2003 poesía 80 integrantes.

El 2 de febrero del 2004, el consorcio anunció la reorganización de Eclipse en una corporación sin ánimo de lucro, independientemente de su fundadora original IBM, denominada Fundación Eclipse. Desde ese momento el código fuente y la tecnología desarrollada por la comunidad de Eclipse estuvo disponible gratuitamente a través de la *Eclipse Public License*. Eclipse se distribuye actualmente bajo la licencia CPL (*Common Public License* o Licencia Publica Común) versión 1.0 de IBM, aprobada por la organización Open Source Initiative (OSI) (27).

Hasta ahora se ha visto el origen del Eclipse pero aún queda la interrogante de lo que es en realidad el Eclipse. La web oficial de Eclipse define al mismo como “una plataforma (IDE), abierta para todo y para nada en particular” (28). Eclipse es una plataforma porque no se encuentra acabada en su totalidad, pero está diseñado para que sea extensible indefinidamente con la adecuada implementación de plug-in. Es un ambiente de desarrollo integrado (IDE) porque provee de herramientas que facilitan el trabajo en el desarrollo de software, porque permite administrar el espacio de trabajo o workspace, porque permite compilar, correr y depurar aplicaciones, porque además posee herramientas que permiten compartir elementos y control de versión sobre el código fuente.

La característica clave de Eclipse es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos plug-ins que van conformando la funcionalidad final. La forma en que los plug-ins interactúan es mediante interfaces o puntos de extensión; así, las nuevas aportaciones se integran sin dificultad ni conflictos. El Eclipse es neutral y apropiado para todo, puesto que ha sido utilizado para desarrollar todo tipo de aplicaciones, y ha sido altamente probado en cualquier ambiente, ya sea en la construcción de Servicios Web, aplicaciones desktop, juegos, etcétera.

- Netbeans

El IDE NetBeans es un entorno integrado de desarrollo galardonado, 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, y de escritorio y aplicaciones móviles utilizando la plataforma Java, así como Java FX, PHP, JavaScript y Ajax, Ruby y Ruby onRails, Groovy y Grails, y C / C + +. El proyecto NetBeans es compatible con una vibrante comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación, así como una amplia selección de plug-ins de terceros. Completo soporte para PHP 5.3: namespaces, funciones lambda y cierres.

Posee además un completo soporte para Symfony, permitiendo generar proyectos y trabajar con las librerías del framework, incluyendo los archivos YAML. Crea un proyecto PHP desde un control remoto de aplicaciones PHP, PHPUnit, la cobertura de código, FTP

/ SFTP y mejoras de integración (29). Para darle solución a la propuesta se usará NetBeans 7.2 por las ventajas que posee en cuanto a completamiento de código, integración con base de datos, además de mantener una estructura en el proyecto creado muy fácil de gestionar.

- Web Services.

Los servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan en colaboración y en las cuales el software está distribuido en diferentes servidores. Los servicios Web XML permiten que las aplicaciones trabajen en conjunto, haciendo uso de funcionalidades brindadas por otras aplicaciones independientemente de cómo se hayan creado, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque los serviciosWeb XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

Los servicios XML WebServices son los elementos fundamentales en la evolución hacia la computación distribuida a través de Internet. Se están convirtiendo en la plataforma de integración de aplicaciones gracias a los estándares abiertos y al énfasis en la comunicación y colaboración entre personas y aplicaciones. Las aplicaciones se crean utilizando los servicios XML WebServices múltiples de origen distinto que funcionan conjuntamente, sin importar su ubicación o la forma en que se implementaron. En la UCI todas las aplicaciones que se utilizan en la intranet ofrecen o “consumen” servicios Web de otras, es decir, existe una interrelación entre los sistemas de la red para lograr la reutilización y la funcionalidad de estas.

- PgAdmin III

Herramienta muy popular y completa de código abierto para administración y desarrollo en la plataforma PostgreSQL, la más avanzada base de datos de código abierto en el mundo. Puede ser usado en Linux, FreeBSD, Solaris, Mac OSX y Windows para administrar PostgreSQL8.3 y anteriores, así como comerciales y las versiones derivadas de PostgreSQL como *Postgres Plus Advanced Server* y *Green plum database*.

Está diseñado para responder a las necesidades de todos los usuarios, de la escritura simples consultas SQL hasta para el desarrollo de bases de datos complejas. La interfaz gráfica soporta todas las características y hace fácil administración. Es desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo y está disponible en más de una docena de idiomas. Es un software libre publicado bajo la licencia BSD[4]. (PgAdmin

[4]**Licencia BSD:** licencia de software libre permisiva, permite el uso del código fuente en software no libre.

PostgreSQL Tools, 2010). Se usará para administrar la base de datos de la solución propuesta, por las ventajas que propone la herramienta.

- Java

Como lenguaje de programación se utilizará Java, este fue desarrollado por la compañía Sun Microsystems. Es un lenguaje de programación orientado a objeto que muchas de sus sintaxis son tomadas de C y C++, pero tiene un modelo de objeto más simple y elimina herramientas de bajo nivel, que por lo general suelen inducir errores como la manipulación directa de punteros o memoria. Es un lenguaje muy extendido y cada vez con más importancia ya que es independiente de la plataforma, no importa el sistema operativo con el que se esté trabajando. Esto se debe al desarrollo de una máquina virtual para cada sistema que sirve de puente entre el sistema operativo y el programa de Java.

Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre. En la actualidad Java se utiliza debido a sus grandes posibilidades y en gran medida a que la mayoría de las cosas que se hacen en cualquier otro lenguaje de programación se puede hacer en Java y muchas veces con grandes ventajas.

Una de las ventajas fundamentales de usar Java es que por ser software libre no es necesario comprar licencias de uso, que generalmente son muy caras, y permite la libre distribución del código sin violar los principios del software libre. Este lenguaje será usado para implementar la aplicación de escucha para los agentes y los agentes que estarán en cada estación de trabajo.

- MySQL

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple o complejas en volumen de información. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. MySQL es software de fuente abierta. Fuente abierta significa que es posible

para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades (30).

- ❖ PostgreSQL

Es un sistema de base de datos relacional de código abierto, que se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Este cuenta con diversas versiones para sistemas operativos tales como: Linux, Windows, Unix, Mac OS X, Solaris, BSD, Tru64 y otros. Soporta vistas, uniones, claves extranjeras, triggers. Incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99 y presenta soporte de protocolo de comunicación encriptado por SSL, extensiones para alta disponibilidad, nuevos tipos de índices y minería de datos.

Permite crear, editar, copiar, extraer y bajar todo objeto de las bases de datos tales como esquemas, tablas, vistas, funciones, dominios, reglas, secuencias, idiomas, operadores, etc. Construye consultas visualmente, ejecuta consultas y scripts SQL, visualiza y edita datos, representa datos como diagramas, exporta e importa datos desde y hacia los formatos de archivos de uso más popular. Por otra parte, administra roles, usuarios, grupos y sus privilegios, y usa una serie de herramientas adicionales diseñadas para una fácil y eficiente operación con el Servidor PostgreSQL.

Dentro de las características que se destacan de PostgreSQL y que lo convierten en una herramienta eficiente para el trabajo en las base de datos, están la atomicidad, la consistencia, el aislamiento y la durabilidad. Juntas aseguran que sólo empieza aquello que se puede acabar y garantizan que una operación no puede afectar a otras. Una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema. El gestor de base de datos PostgeSQL será usado por las ventajas de ser libre, rápido y su durabilidad.

- ❖ CodeIgniter

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación Web usando PHP. Su objetivo es permitir al desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas más comunes, así como una interface simple y estructura lógica para acceder a esas librerías. CodeIgniter gestiona la seguridad de diferentes formas. Es justamente restrictivo sobre que caracteres permitir en las cadenas URI[5] para ayudar a minimizar la posibilidad de que datos maliciosos puedan ser pasados a su aplicación (31). Las URL sólo pueden contener lo siguiente:

[5] URI: Identificador de Recurso Uniforme (*Uniform Resource Identifiers* por sus siglas en inglés).

- Texto alfanumérico.
- Tilde.
- Punto.
- Dos puntos.
- Guión bajo.
- Guión.

Los datos GET son simplemente anulados por CodeIgniter ya que el sistema utiliza segmentos URI en vez de las tradicionales *query string* de URL (a menos que la opción query string esté habilitada en su archivo de configuración). El arreglo global GET es destruido por la clase de Entrada (input) durante la inicialización del sistema. Desde la inicialización del sistema, todas las variables globales son destruidas, excepto aquellas que son encontradas en los arreglos $_POST y $_COOKIE. La rutina de eliminación es efectivamente la misma que register_globals = off.

La directiva *magic_quotes_runtime* es apagada durante la inicialización del sistema para que no tenga que remover las barras cuando se recuperen datos de la base de datos. CodeIgniter viene con un filtro de Cross Site Scripting (XSS). Este filtro busca técnicas comúnmente usadas para incluir Java script malicioso a sus datos, u otro tipo de código que intente secuestrar cookies.

CodeIgniter solo implementa defensas contra algunos tipos de ataque mediante funciones que le permiten tener direcciones URL más segura y mediante el tratamiento de algunas variables propias el PHP, lo que hace engorrosa la configuración. No posee un sistema de control de acceso definido, sino que estos mecanismos tienen que ser implementados por el propio desarrollador. Por sus características y ventajas la herramienta será usada como framework *server-side* de la solución propuesta.

- ExtJS

La programación con la librería ExtJS, que extiende la librería YUI e integra AJAX, Prototype y Scriptaculous. Con el tiempo se convirtió en un framework independiente y a principio de 2007 se creó una compañía para comercializar y dar soporte al ExtJS.ExtJS es neutral al lenguaje que se use en el servidor. Siempre que el resultado se envíe a la página en el formato adecuado, ExtJS no se preocupará de lo que pase en el servidor. Hay docenas de widgets a escoger en ExtJS, incluyendo composiciones automáticas de páginas, pestañas, menús, barras de herramientas, diálogos, vistas en árbol.

Proporciona un selector de nodos DOM extremadamente poderoso llamado DomQuery (puede usarse como una librería independiente, pero en el contexto de ExtJS se usará para seleccionar elementos para poder interactuar con ellos a través de la interfaz

Element, contiene mucho de los métodos y propiedades de DOM que se necesitará proporcionando una interfaz conveniente, unificada y multi-navegador). (32)

Para trabajar con las librerías de ExtJS es necesario que exista un adaptador, donde esta clase sobre la que luego se define las funciones de las librería (el elemento Ext como tal). A partir de la versión 1.1, ExtJS proporciona su propio adaptador, ya no es necesario incluir el de YUI, o JQuery o Scriptaculous (incluir en el caso de usar alguna función de estas librerías que no esté en Ext) En su última versión Ext 2.0 API es muy extenso y recordando todas las funciones, propiedades o configuraciones disponible es casi imposible. La documentación del API está muy completa.

Se emplean IDEs de desarrollo de JavaScript, el Aptana Studio es una herramienta potente y disponible para el apoyo directo en las aplicaciones desarrolladas en ExtJS. Por sus ventajas y características la tecnología será usada como framework *client-side* de la solución propuesta, posee la ventaja de diseñar con componentes una interfaz de usuario reutilizables y brinda un patrón modelo-vista-controlador.

1.8 Metodologías para el desarrollo de la solución propuesta

- RUP

Es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El proceso de software propuesto por RUP tiene tres características esenciales: Guiado por los casos de uso: Los casos de uso no son solo una herramienta para especificar los requisitos del sistema sino que constituyen un elemento integrador y una guía del trabajo, de su diseño, implementación y prueba.

Además, estos no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo. Centrado en la arquitectura: La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. (33). Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones.

Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Dentro de los flujos de trabajo del

proceso de desarrollo se encuentran el Modelado de Negocio, la Gestión de Requerimientos, Análisis y Diseño entre otros, la particularidad de todos es que hacen uso del Lenguaje Unificado de modelado para representar los artefactos que se generan en cada uno de ellos.

- XP

La Programación Extrema (XP) surge ideada por Kent Beck, como proceso de creación de software diferente al convencional. En palabras de Beck: "XP es una metodología ligera, eficiente, con bajo riesgo, flexible, predecible y divertida para desarrollar software". XP es una de las metodologías ágiles para el desarrollo de software más exitosas de la actualidad. Se utiliza en proyectos con pequeños equipos de desarrollo y con corto plazo de entrega. Se basa en la retroalimentación entre el cliente y el equipo de desarrollo, buena comunicación entre los participantes y simplicidad en las soluciones implementadas.

Consiste en una programación rápida, cuya particularidad es que tiene como miembro del equipo al usuario final. Es adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (34). En XP se sigue la idea de la programación en pares, dado las ventajas que ofrece respecto a la creación del código, pues se pueden evitar errores y malos diseños al controlar cada línea de código y decisión de diseño instantáneamente (35). La interacción entre ambos desarrolladores puede generar discusiones que lleven a mejores estructuras y algoritmos, aumentando la calidad del software.

- OpenUp

Es una metodología de desarrollo basada en la metodología RUP (*Rational Unified Process*). Es el subconjunto de esta última que contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo de software a realizar un producto de alta calidad y de una forma más eficiente. Utiliza un punto de vista pragmático y una filosofía ágil que se centraliza en la naturaleza colaborativa del proceso de desarrollo del software. Una de sus principales características es su alto grado de adaptabilidad a las necesidades de un proyecto en particular.

OpenUP es un proceso de desarrollo iterativo del software que es mínimo, completo, y extensible. Como metodología de desarrollo es conducida por el principio de colaboración para alinear intereses y para compartir su comprensión. Es el proceso unificado que aplica acercamientos iterativos e incrementales dentro de un ciclo vital estructurado (36).

Se identificó la metodología OpenUp como la metodología más adecuada para el desarrollo de la solución propuesta por las siguientes características y ventajas:

- Es completo en el sentido que puede ser manifestado como todo el proceso para construir un sistema.
- Es extensible ya que el proceso se pueda agregar o adaptar según lo vayan requiriendo los sistemas.
- OpenUP es un proceso ágil.
- Es ligero y proporciona una comprensión detallada del proyecto, beneficiando a clientes y desarrolladores sobre el producto a entregar y su formalidad.
- Se centra en una arquitectura temprana para reducir al mínimo los riesgos y organizar el desarrollo.
- OpenUp es la metodología utilizada por desarrolladores de alto nivel en casi todo el mundo por sus altas cualidades administrativas.

OpenUp tiene las características del Proceso Unificado (*Unified Process*) al cual se aplican enfoques iterativos e incrementales dentro de un ciclo de vida estructurado. El proceso de desarrollo utiliza casos de uso, escenarios, gestión del riesgo y un enfoque centralizado en la arquitectura (37).

El objetivo de esta metodología es ayudar al equipo de desarrollo a través de todo el ciclo de vida de las iteraciones, de modo que este sea capaz de añadir valor de negocio para los clientes de una forma predecible: con la entrega de un software operativo y funcional al final de cada iteración. El ciclo de vida del proyecto provee a los clientes de una visión del proyecto, transparencia y les dota de los medios para que puedan controlar la financiación, el riesgo, el ámbito, el valor de retorno esperado, etc.

Conclusiones

Después de haber realizado un estudio acerca de los sistemas existentes a nivel nacional e internacional, y de determinar los principales elementos que los caracterizan, se definieron los posibles aspectos a tener en cuenta para desarrollar la propuesta de solución, dentro de los cuales se encuentra mantener la tecnología que se aplica en la Universidad (código de barras). Esta tecnología es mucho más económica, pero no se puede dejar de reconocer las ventajas que pudiera traer para la Universidad la utilización de un equipamiento que pueda hacer mucho más eficiente los procesos que se realizan.

Por otra parte se realizó una fundamentación y selección de las herramientas y tecnologías a utilizar en el análisis, diseño e implementación de la aplicación y además se describieron las principales características de la metodología seleccionada para la solución propuesta.

Capítulo 2 Características de la aplicación

Introducción

Después de analizar los diferentes puntos teóricos acerca de la investigación con respecto a los controladores de acceso, las herramientas y metodología que se utilizarán en el desarrollo de la aplicación. Se comenzará con una breve descripción del negocio, el modelado del negocio correspondiente, identificándose los actores, trabajadores y los casos de uso, además del modelo de objetos todo esto conllevará a un mejor entendimiento de los procesos a informatizar.

Esta capitulo abordará las características de la aplicación que se pretende desarrollar, se definen además los requisitos funcionales y no funcionales, el diagrama de casos de uso del sistema y la descripción de cada caso de uso del sistema correspondiente. El proceso de desarrollo a seguir estará guiado por las pautas que se rige la metodología de desarrollo OpenUp.

2.1 Descripción del sistema propuesto

Para solucionar los problemas existentes en la universidad se propone realizar un sistema informático, cuyo objetivo sea elevar la calidad de todos los procesos que se desarrollen en los laboratorios de la UCI en cuanto a control de acceso. El sistema permitirá establecer un control de las entradas y salidas del personal que accede a los laboratorios, el acceso a las estaciones de trabajo en cuanto a permisos sobre las sesiones, restringir o permitir las aplicaciones que se activan en las estaciones de trabajo y la generación de reportes a partir de los informes de los agentes activos en las estaciones de trabajo.

Este sistema contendrá varios servicios automatizando todo el proceso de acceso a los laboratorios y presentará una mayor fuente de información para la toma de decisiones administrativas, así como un mayor control sobre la tecnología y su utilización en el área de los laboratorios.

2.2 Descripción del negocio

La universidad de las ciencias informáticas (UCI) cuenta con el área de los laboratorios, pues entre sus objetivos principales se encuentra la formación de estudiantes y profesores en cuanto a docencia e investigación, la asistencia de los mismos a dichos laboratorios es sistemática debido a las tareas orientadas por la docencia y los proyectos. Las personas que pueden acceder al laboratorio están reflejadas en una lista que es creada por el jefe de departamento, responsable máximo del laboratorio y sus funciones, la lista es entregada al técnico de laboratorio. En caso de que el usuario no tenga acceso puede solicitar al jefe de departamento el acceso a un laboratorio, si este verifica que el usuario pertenece al laboratorio entonces este actualiza la lista de usuarios con acceso y

se la entrega al técnico de dicho laboratorio, el técnico procederá a asignar una estación de trabajo al usuario.

El técnico de laboratorio posee varias funciones específicas y entre ellas es controlar que cada persona que solicite el acceso al laboratorio sea exactamente de esa facultad, centro y departamento y si pertenece a ese proyecto usando la lista que le proporcionó el jefe de departamento. Si el usuario pertenece a ese laboratorio, el técnico procede a recoger el nombre y apellidos, solapín, laboratorio además de la fecha y hora en que ocurrió la incidencia. En caso contrario de que no pertenezca a ese laboratorio, se le comunica y el usuario procede a la salida del laboratorio.

El técnico después de asignar la estación de trabajo al usuario, debe controlar que el mismo cumpla con las directivas de seguridad informática, no se debe jugar juegos de video, ver series o películas, evitando un maltrato de la tecnología disponible, en caso de encontrar algún usuario en algunas de estas funciones, se le llamara la atención y se le pedirá el abandono del laboratorio, tomando sus datos para un posterior análisis con el jefe de departamento.

Una vez que el usuario haya terminado su tiempo de trabajo en la estación de trabajo y no sean violadas ninguna de las directivas de seguridad informática, procede a apagar el equipo y el técnico recogerá la fecha y la hora que este salió del laboratorio. El técnico al finalizar su labor debe entregar al jefe de departamento la lista con todos los accesos e incidencias recogidas, esta información servirá para la toma de decisiones en cuanto a utilización y disponibilidad de la tecnología, tiempo de utilización de los laboratorios y sus estaciones de trabajo, entre otras.

2.3 Descripción de los actores y trabajadores del negocio

Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. Los actores representan terceros fuera del sistema que colaboran con el sistema. Una vez que tengamos identificados todos los actores del sistema, tenemos identificado el entorno externo del sistema. (38)

Tabla 1 Actores del Negocio

Nombre del actor	Descripción
Usuario	Es el que solicita y se beneficia con el servicio de los laboratorios de la UCI.

Trabajadores del negocio

Un trabajador define el comportamiento y las responsabilidades de un individuo que actúa en el negocio realizando una o varias actividades, interactuando con los actores del negocio y manipulando entidades del negocio.

Tabla 2 Trabajadores del Negocio

Trabajador	Descripción
Técnico de laboratorio	Persona responsable del laboratorio y de hacer cumplir con el reglamento, directivas de seguridad informática y disciplina dentro del mismo.
Jefe de departamento	Persona que evalúa y supervisa los técnicos, además de crear la lista de accesos a los laboratorios.

2.4 Descripción de casos de uso del negocio

Caso de Uso del negocio: “Entrar al laboratorio”.

Caso de uso:	Entrar al laboratorio
Actores:	Usuario (inicia)
Trabajadores:	Técnico de laboratorio
Resumen:	El caso de uso se inicia cuando el usuario entra al laboratorio solicitando al técnico de laboratorio el uso de una estación de trabajo. El técnico verifica si pertenece a ese laboratorio, en caso de ser así, le toma los datos de nombre, apellidos, solapín, fecha y hora, de existir en la lista de acceso le permite acceder al laboratorio y a una estación de trabajo. De no pertenecer al laboratorio, el usuario se retira del laboratorio.

Caso de Uso del negocio: “Usar estación de trabajo”.

Caso de uso:	Usar estación de trabajo
Actores:	Usuario (inicia)
Trabajadores:	Técnico de laboratorio
Resumen:	El caso de uso se inicia cuando el usuario tiene permiso para usar la estación de trabajo, el usuario comienza a usar la

	estación de trabajo y el técnico verifica cada cierto tiempo que el usuario cumpla todas las directivas de seguridad dentro del laboratorio, de no cumplirlas el técnico retirará al usuario de la estación de trabajo, y este último abandonara el laboratorio.

Caso de Uso del negocio: “Solicitar acceso al laboratorio”.

Caso de uso:	Solicitar acceso al laboratorio
Actores:	Usuario (inicia)
Trabajadores:	Jefe de departamento
Resumen:	El caso de uso se inicia cuando el usuario solicita al jefe de departamento el acceso a un laboratorio específico, si este lo considera correcto, entonces actualiza la lista de accesos para posteriormente entregarla al técnico de laboratorio.

Caso de Uso del negocio: “Salir del laboratorio”.

Caso de uso:	Salir del laboratorio
Actores:	Usuario (inicia)
Trabajadores:	Técnico de laboratorio
Resumen:	El caso de uso se inicia cuando el usuario desea retirarse y pasado un tiempo de uso en la estación de trabajo, se dirige al técnico, el cual le toma los datos: nombre, apellidos, solapín, fecha y hora de salida, el usuario sale del laboratorio.

A continuación presentamos el diagrama de casos de uso del negocio, lo que nos facilitará una mejor compresión de la relación entre el actor y los casos de uso del negocio.

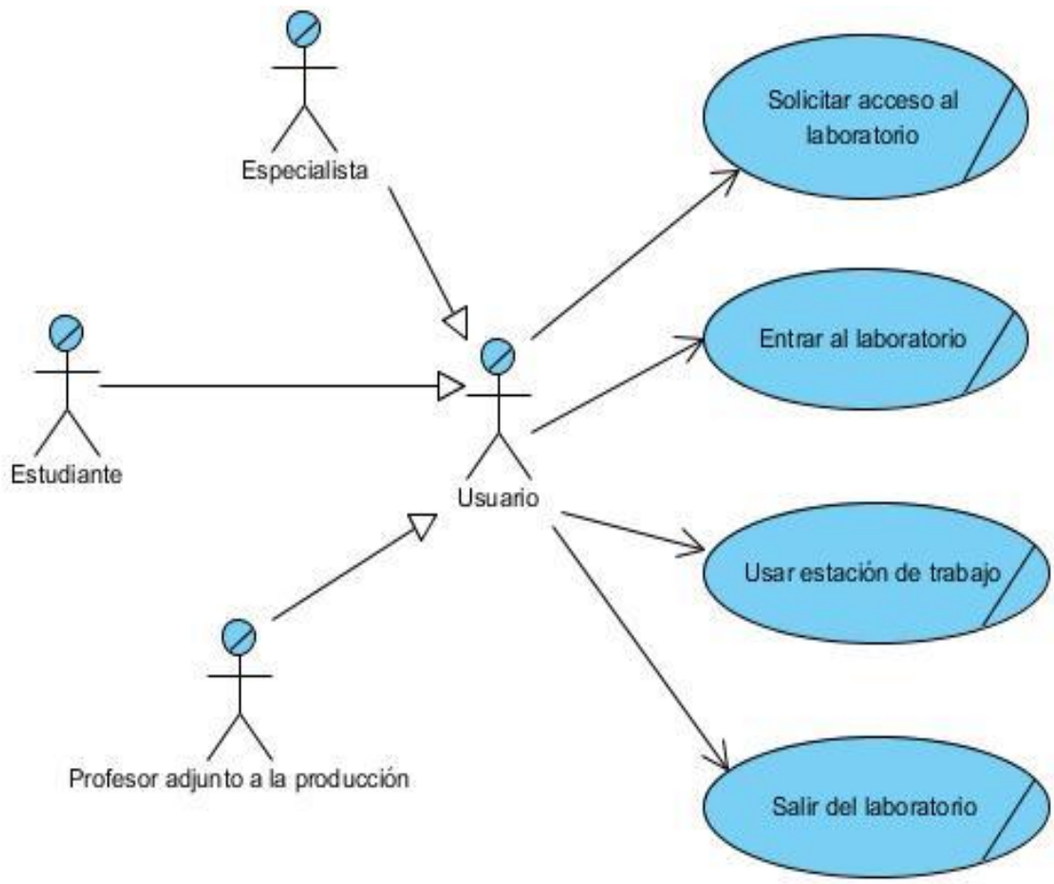


Figura 1: Diagrama de casos de uso del negocio

2.5 Diagrama de objetos del negocio

Un modelo de objetos del negocio es un modelo interno a un negocio. El mismo describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio.

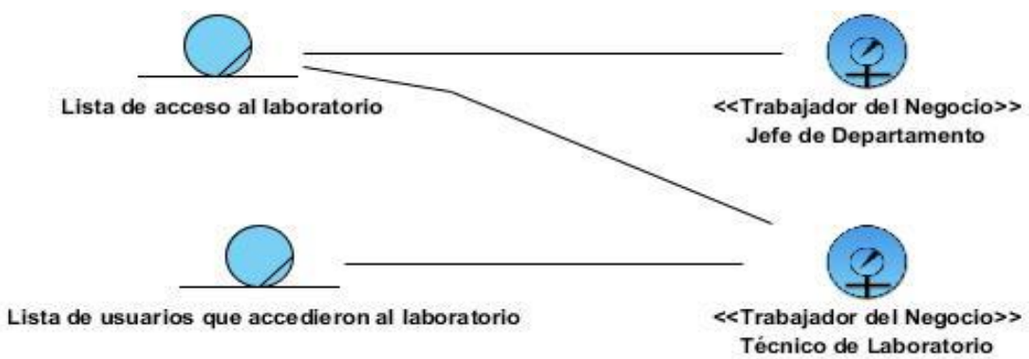


Figura 2: Diagrama de Objetos

2.6 Diagramas de actividades de casos de uso

Un diagrama de actividades es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro del negocio. Donde participan los actores/trabajadores del negocio, las actividades a realizar por ellos y las relaciones entre estas. Se coloreara de rojo las actividades que se informatizarán, a partir de estas actividades se extraerán los requisitos:

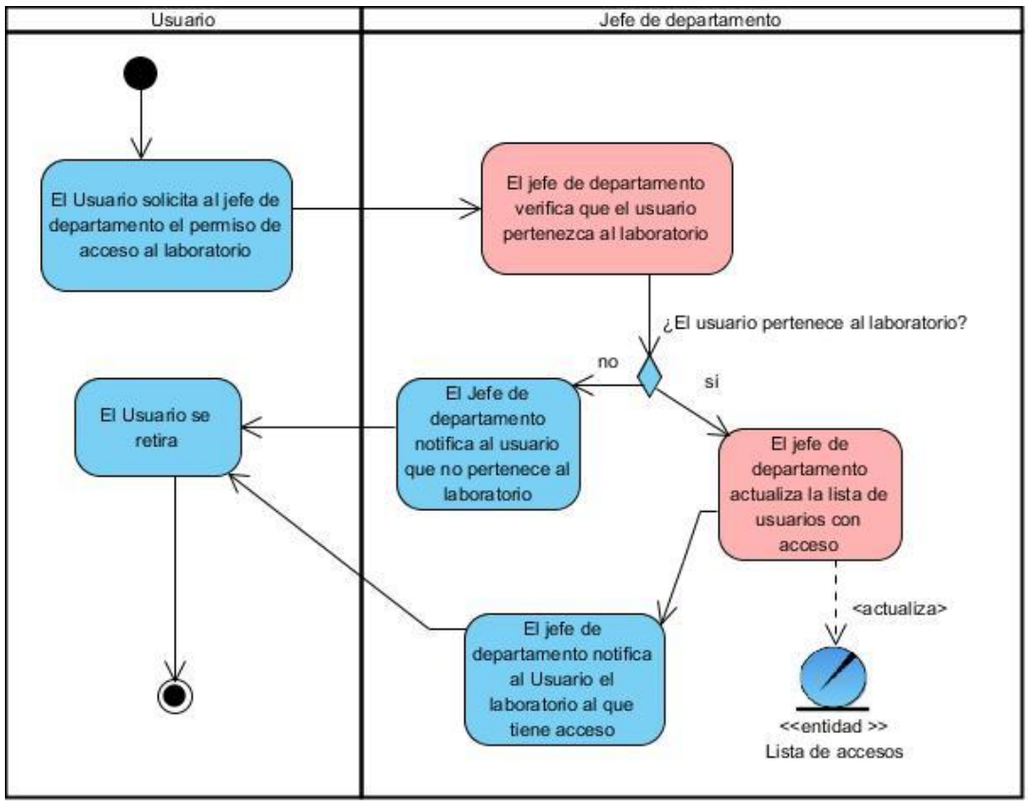


Figura 3: Diagrama de Actividades para el CUN "Solicitar acceso al laboratorio"

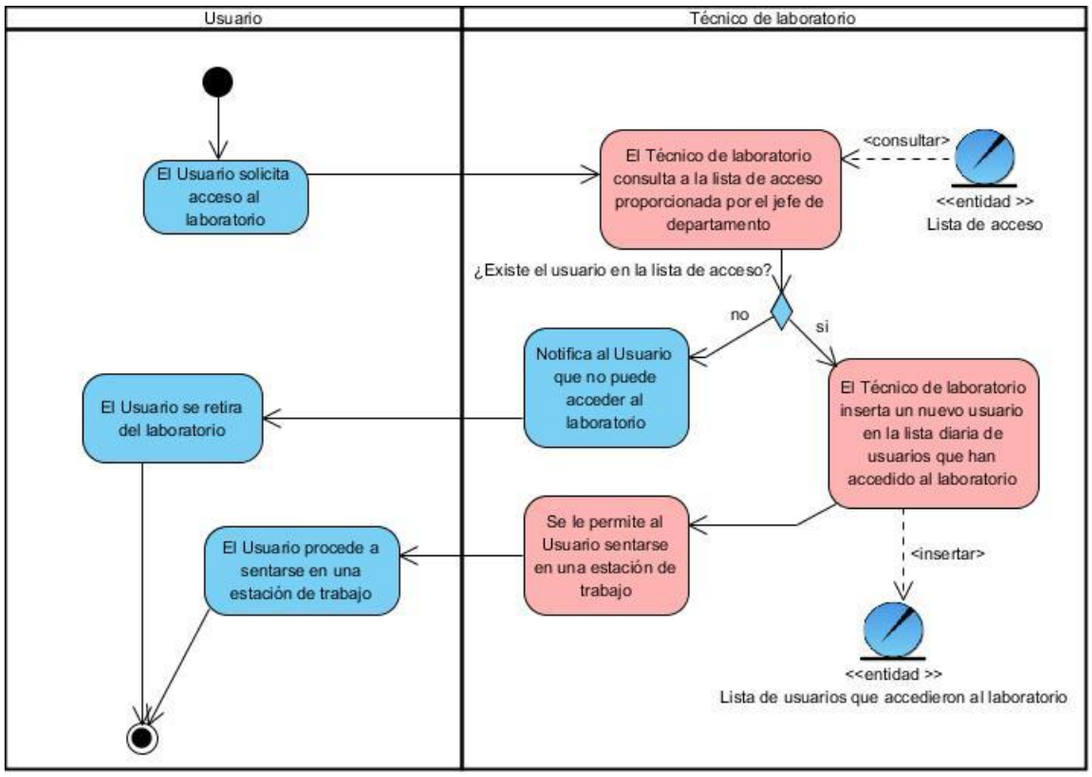


Figura 4: Diagrama de Actividades para el CUN "Entrar al laboratorio"

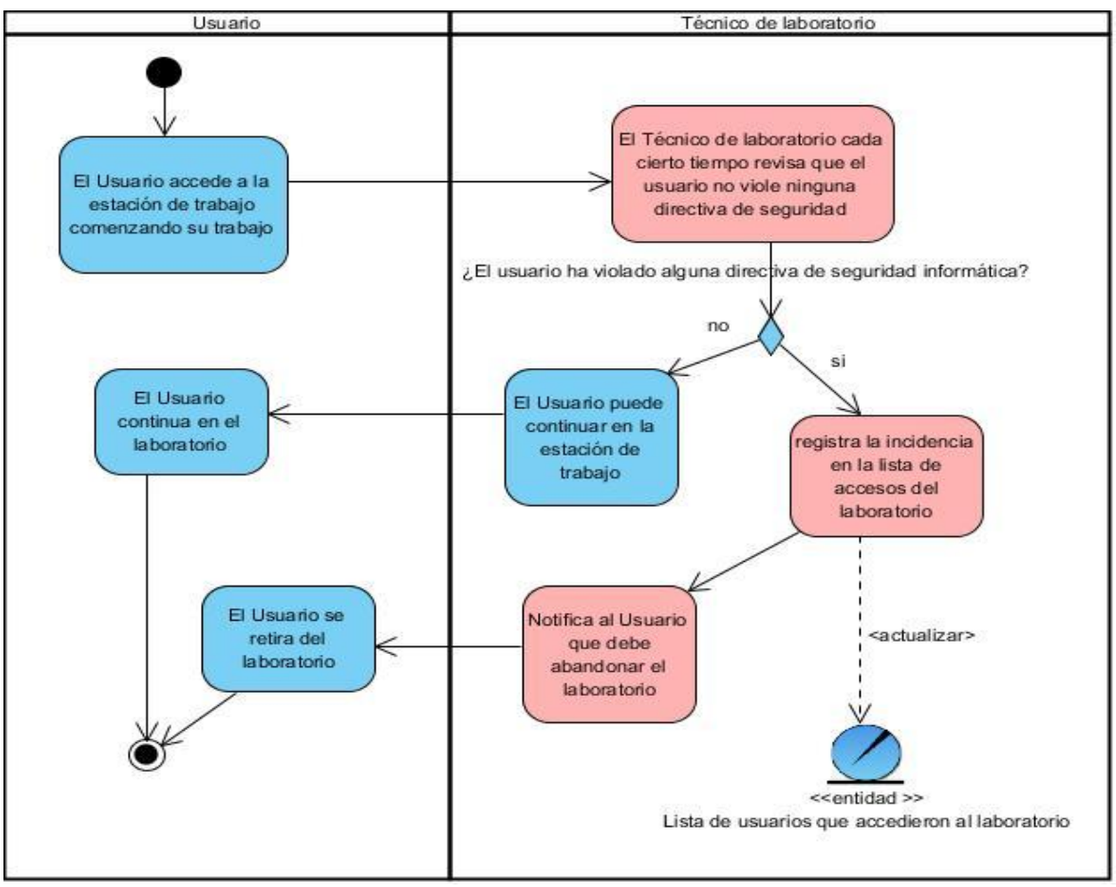


Figura 5: Diagrama de Actividades para el CUN "Usar estación de trabajo"

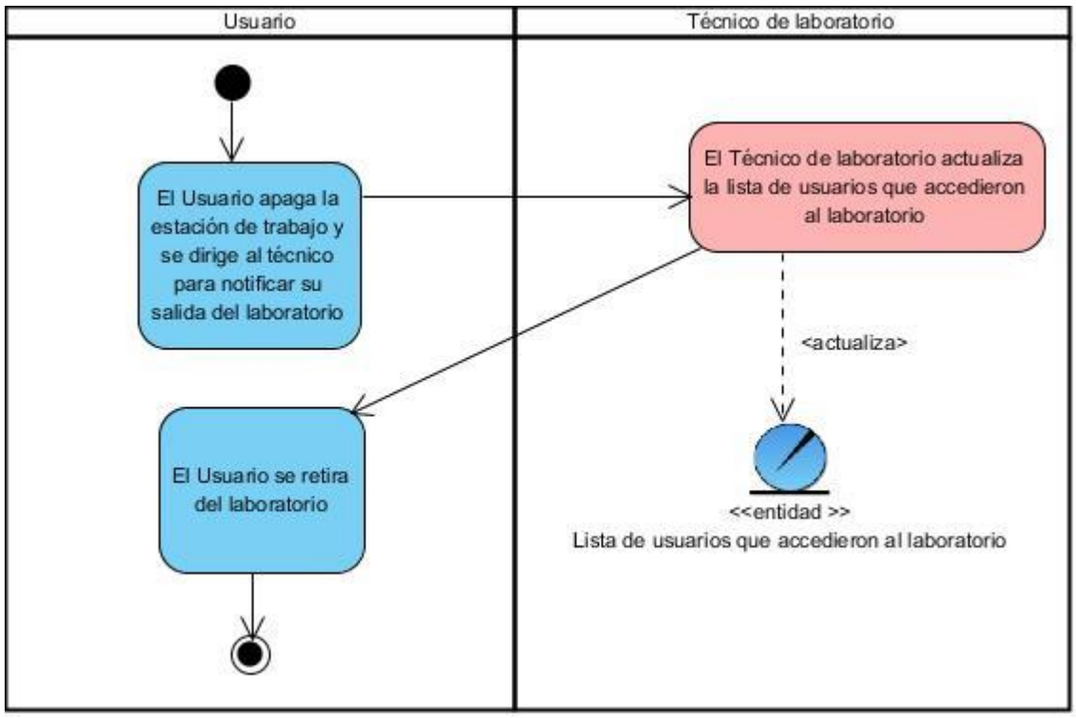


Figura 6: Diagrama de Actividades para el CUN "Salir del laboratorio"

2.7 Requerimientos

Un requerimiento es una condición o capacidad necesaria para que un usuario resuelva un problema o alcance un objetivo. A continuación se describen los requerimientos funcionales y no funcionales del sistema.

2.7.1 Requerimientos funcionales

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. Los requerimientos para un sistema de software determinan lo que hará el sistema y definen las restricciones de su operación e implementación. Para cumplir con los objetivos propuestos se plantean las siguientes funcionalidades:

RF 1. Autenticar usuario.
RF 2. Crear entrada y salida de usuarios al laboratorio.
RF 3. Eliminar entrada y salida de usuarios al laboratorio.
RF 4. Crear sesión de computador desde estación de trabajo.
RF 5. Verificar estado de sesión en sistema operativo.
RF 6. Verificar existencia de servidor de agentes.
RF 7. Verificar existencia de sesión en agente.

RF 8. Verificar el estado del usuario.
RF 9. Obtener procesos de estación de trabajo en agente.
RF 10. Bloquear proceso.
RF 11. Eliminar proceso.
RF 12. Salvar procesos y usuarios localmente.
RF 13. Procesar petición de agente.
RF 14. Insertar facultad
RF 15. Actualizar facultad
RF 16. Eliminar facultad
RF 17. Insertar centro
RF 18. Actualizar centro
RF 19. Eliminar centro
RF 20. Insertar departamento
RF 21. Actualizar departamento
RF 22. Eliminar departamento
RF 23. Insertar laboratorio
RF 24. Actualizar laboratorio
RF 25. Eliminar laboratorio
RF 26. Insertar proyecto
RF 27. Insertar personas a proyecto
RF 28. Actualizar proyecto
RF 29. Eliminar proyecto
RF 30. Insertar computador
RF 31. Actualizar computador
RF 32. Eliminar computador
RF 33. Adicionar proceso
RF 34. Bloquear proceso
RF 35. Eliminar proceso
RF 36. Adicionar persona
RF 37. Actualizar persona
RF 38. Eliminar persona
RF 39. Bloquear persona
RF 40. Anclar persona
RF 41. Eliminar sesión de computador
RF 42. Eliminar sesión de agente
RF 43. Eliminar proceso de agente según sesión
RF 44. Insertar rol
RF 45. Actualizar rol

RF 46. Eliminar rol
RF 47. Insertar permiso
RF 48. Actualizar permiso
RF 49. Eliminar permiso
RF 50. Configurar reporte
RF 51. Generar reporte.

2.7.2 Requerimientos no funcionales

Llámese requisito no funcional a todas las exigencias de cualidades que se le imponen al proyecto: exigencias de usar un cierto lenguaje de programación o plataforma tecnológica por ejemplo. Es una característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo.

Aseguran que se disponga de un sistema manejable y gestionable que ofrezca la funcionalidad requerida de manera fiable, ininterrumpida o con el tiempo mínimo de interrupción, incluso ante situaciones inusuales.

Requerimientos de apariencia o interfaz externa: El sistema debe tener un ambiente amigable y entendible para los usuarios finales de forma tal que no les sea muy complicado utilizar el software.

Requerimientos de portabilidad: Debe tener facilidad para adaptarlo a diferentes ambientes. Independencia de la plataforma.

Requerimientos de usabilidad: El sistema está destinado principalmente a los directivos que desean controlar el acceso a los laboratorios, de igual forma puede ser utilizado por cualquier otro individuo. Los usuarios finales deberán poseer conocimientos básicos de computación, navegación y exploración de los sitios web en sentido general para lograr un mejor uso de las funcionalidades que brinda la aplicación.

Requerimientos de software: Para obtener un óptimo funcionamiento del sistema se requiere la existencia de los siguientes requisitos en el servidor y las maquinas clientes que harán uso de la aplicación:

Cliente:

- Cualquier sistema operativo con interfaz gráfica.
- Red activa.
- Cualquier navegador web.
- Máquina Virtual de Java (JVM) versión 6.0 o mayor.

Servidor

- Sistema operativo Linux en cualquiera de sus distribuciones.
- Servidor de base de datos PostgreSQL 9.1
- Servidor web Apache 2.26
- Máquina Virtual de Java (JVM) versión 6.0 o mayor
- Lenguaje PHP 5.3.3

Requerimientos de seguridad y privacidad: Fueron definidas un grupo de políticas de seguridad que prevén el uso inadecuado del sistema. Los usuarios deberán identificarse antes de acceder a cualquier acción del sistema, así se garantiza que la aplicación sea utilizada solo por aquellos usuarios que tienen permisos. El administrador es el máximo responsable de la aplicación y el encargado de asignar estos permisos. En el sistema se verificarán las acciones irreversibles antes de ejecutarse como mecanismo de seguridad y se asegura de que el sistema funcione correctamente aún cuando no haya conectividad.

La contraseña de los usuarios del dominio es manipulada con minucioso cuidado por su alto grado de confiabilidad que representa dentro de la UCI, utilizando para ello el servicio WSDL de autenticación que brinda la Universidad, evitando que dicha información sea almacenada en el Servidor Web que alberga la aplicación. Para el resto de las contraseñas se utiliza BD con restricciones de seguridad previamente configuradas y mecanismos de encriptación (MD5) evitando que la información viaje en texto plano (claro).

Requerimientos de hardware: El buen estado de las conexiones de red de la Universidad es imprescindible y de suma importancia para el control de las estaciones de trabajo mediante los agentes, para el envío de información al servidor de agentes y su posterior consulta.

Cliente

- Tarjeta de red de 10 Megabytes (Mb) o superior.
- 128 Mb de RAM

Servidor

- Microprocesador Pentium 4
- Memoria RAM 2.0 Gigabytes (Gb)
- Capacidad de disco duro 80 Gb.

2.8 Actores y casos de uso del sistema

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, no son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información.

Un caso de uso no es más que una secuencia de actividades que realiza un sistema y que da como resultado un valor para el actor. Estos han alcanzado un uso universal debido a dos razones básicas, la primera de ellas es que proporcionan un medio intuitivo y sistemático de capturar los requisitos anteriormente mencionados, centrándose en lo que quiere obtener el cliente, y la segunda es que dirigen todo el proceso apreciando que el análisis, diseño y prueba se realizan partiendo de los casos de uso. Es de vital importancia realizar una buena selección de los casos de uso debido que el proceso de desarrollo está guiado por ellos, lo que se traduce en que, una serie de flujos de trabajo se inicia a partir de los mismos.

2.8.1 Descripción de los actores del sistema

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información. En este caso los actores que interactúan con el sistema se definen en la siguiente tabla.

Actor	Descripción
Usuario	Representa al actor encargado de interactuar con el sistema para beneficiarse de las funcionalidades.
Técnico de laboratorio	Representa al actor encargado de tomar los datos del usuario en la entrada del laboratorio, además puede generar reportes.
Jefe de departamento	Representa la máxima expresión de autoridad en el sistema, administrar completamente el sistema y generar reportes.
Usuario JKEEPER	Representa a los usuarios que se autenticarán en el sistema, configurarán y generarán los reportes.

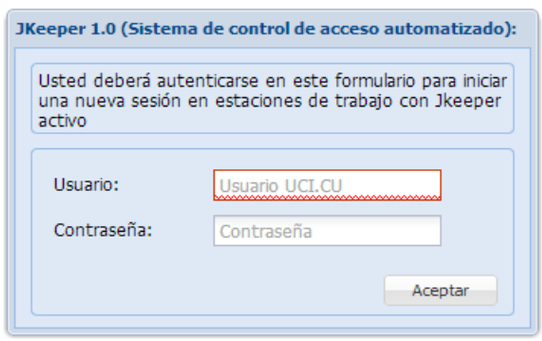
2.8.2 Descripción de los casos de uso del sistema

1. Caso de uso “Crear sesión de computador”

Caso de Uso:	Crear sesión de computador
Actores:	Usuario
Resumen:	El caso de uso del sistema se inicia cuando el usuario inicia su sesión de dominio en la estación de trabajo, el usuario deberá iniciar el navegador y autenticarse en el sistema, creando una sesión de computador para su posterior utilización.
Prioridad:	Alta
Precondiciones:	Inicio de sesión en el sistema operativo.

Flujo normal de eventos	
Acción del actor:	**Respuesta del sistema**
1. El caso de uso se inicia cuando el usuario inicia sesión en el sistema operativo.	2. Se muestra la interfaz de usuario que contiene los campos Usuario y Contraseña.
3. El usuario entra los datos de Usuario y Contraseña.	4. Se verifica si el usuario existe en la base de datos, en caso contrario (Ver FA 1). 5. Se verifica que el usuario y la contraseña sean auténticos del dominio uci.cu, en caso contrario (Ver FA2). 6. Se verifica que no se haya creado anteriormente otra sesión de computador en la estación de trabajo, en caso contrario (Ver FA3). 7. Se verifica que se haya creado una entrada en el sistema al laboratorio *(ver CU "Crear entrada y salida de usuarios")* donde existe la estación de trabajo. (Ver FA4).

Flujos alternos	
Acción del actor:	**Respuesta del sistema:**
FA1.	Se muestra un mensaje de error indicando que el usuario no se encuentra en la base de datos del sistema, se termina el caso de uso.

FA2.	Se muestra un mensaje de error indicando que el usuario no existe en el dominio o la contraseña esta incorrecta, se termina el caso de uso.
FA3.	Se muestra un mensaje de error indicando que el usuario ha creado previamente una sesión de computador para esta estación de trabajo, se termina el caso de uso.
FA4.	Se muestra un mensaje de error indicando al usuario que debe registrarse en el módulo de acceso del sistema, se termina el caso de uso.
Poscondiciones:	Se creó una nueva sesión de computador, el agente comienza a funcionar correctamente.
Prototipo de interfaz	
	

2. Caso de uso “Verificar existencia de sesión de computador”

Caso de Uso:	Verificar existencia de sesión de computador.
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario inicia sesión o cierra en el sistema operativo de la estación de trabajo
Prioridad:	Alta
Precondiciones:	Inicio de sesión en el sistema operativo o cierre de sesión en el sistema operativo.
Flujo normal de eventos	
Acción del actor:	**Respuesta del sistema**
1. El usuario inicia sesión o cierra la sesión en el sistema operativo de la estación de	2. Si el usuario ha iniciado la sesión en el sistema operativo, el sistema solicita una

trabajo.	sesión de computador existente de no ser así FA1, si existe, el sistema guarda los datos del inicio de sesión. 3. Si el usuario ha cerrado la sesión en el sistema operativo, el sistema cierra la sesión computador abierta y guarda los datos de cierre de sesión.
Flujos alternos	
Acción del actor:	**Respuesta del sistema:**
FA1.	El sistema notifica al usuario que debe crear una sesión de computador en el sistema y cierra la sesión, se termina el caso de uso.
Poscondiciones:	El usuario tiene acceso a la estación de trabajo.

3. Caso de uso “Cerrar proceso en el sistema operativo”

Caso de Uso:	Cerrar proceso en el sistema operativo
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario se encuentra en una estación de trabajo, e inicia un proceso penalizado por el sistema.
Prioridad:	Alta
Precondiciones:	El Usuario debe estar autenticado en el sistema operativo y debe tener una sesión de computador en el sistema.
Flujo normal de eventos	
Acción del actor:	**Respuesta del sistema**
1. El Usuario se encuentra en una estación de trabajo e inicia un proceso bloqueado por el sistema.	2. El sistema reconoce los procesos. 3. Verifica si existe algún proceso penalizado, en caso de ser así, ver FA1
Flujos alternos	
Acción del actor:	**Respuesta del sistema:**
FA1	1. Notifica al usuario que ha incurrido en

	una infracción de seguridad. 2. Cierra el proceso penalizado del sistema operativo.
Poscondiciones:	La sesión del sistema operativo ha sido cerrada y el usuario bloqueado en el sistema.

4. Caso de uso “Controlar acceso de usuarios al laboratorio”

Caso de Uso:	Controlar acceso de usuarios al laboratorio
Actores:	Técnico de laboratorio
Resumen:	El caso de uso se inicia cuando el usuario accede al laboratorio.
Prioridad:	Alta
Precondiciones:	El usuario deberá acceder al laboratorio
Flujo normal de eventos	
Acción del actor:	**Respuesta del sistema**
1. El Técnico de laboratorio introduce el solapín y el laboratorio a donde accederá el usuario.	2. El sistema verifica que el solapín exista en la base de datos, de lo contrario (Ver FA1). 3. El sistema verifica que el usuario con el solapín indicado pertenezca al laboratorio, en caso contrario (Ver FA2) 4. El sistema guarda el acceso del usuario y el laboratorio. 5. El sistema muestra datos de interés para el Técnico de laboratorio.
Flujos alternos	
Acción del actor:	**Respuesta del sistema:**
FA1.	El sistema notifica que el solapín entrado no existe en la base de datos.
FA2.	El sistema notifica que el usuario con ese solapín no pertenece al laboratorio especificado.
Poscondiciones:	Se ha guardado una entrada o salida de un usuario, el usuario puede proceder a usar la estación de trabajo.

5. Caso de uso “Bloquear proceso del sistema operativo”

Caso de Uso:	Bloquear proceso del sistema operativo
Actores:	Jefe de departamento
Resumen:	El caso de uso se inicia cuando el jefe de departamento solicita bloquear un proceso de sistema operativo.
Prioridad:	Alta
Precondiciones:	El Jefe de departamento debe estar autenticado en el sistema.
Flujo normal de eventos	
Acción del actor:	**Respuesta del sistema**
1. El jefe de departamento selecciona el módulo administración, y el sub-módulo procesos. 3. Selecciona bloquear proceso de sistema operativo. 5. El jefe de departamento llena el campo “Motivos del bloqueado”	2. Se listan todos los procesos detectados por el sistema en las estaciones de trabajo. 4. Se muestra una ventana, con un campo de descripción, donde debe entrar los motivos del bloqueado. 6. Se bloquea el proceso
Flujos alternos	
Poscondiciones:	Se ha bloqueado un nuevo proceso en las estaciones de trabajo.

2.9 Diagrama de casos de uso del sistema

El diagrama de casos de uso (DCU) muestra los actores y casos de uso definidos para el sistema propuesto, así como las diferentes relaciones que existen. Para la realización del diagrama se puso en práctica algunos patrones como son CRUD y Actor múltiple. (Ver Anexo 1).

2.10 Patrones de casos de uso

Un patrón de casos de uso define los comportamientos que deben existir en el sistema, ayuda a describir qué es lo que el sistema debe hacer, es decir, describe el uso del sistema y cómo este interactúa con los usuarios. Son utilizados generalmente como plantillas que especifican como deberían ser estructurados y organizados los casos de uso y capturan mejores prácticas para modelar casos de uso.

Los patrones de casos de uso brindan beneficios tales como:

- Aumentar la productividad.
- Reutilizar elementos existentes.
- No invertir tiempo en resolver problemas ya resueltos.
- Aplicar la teoría al trabajo práctico.
- Habilitar las herramientas de soporte para modelar el desarrollo.

Durante el diseño de los casos de uso del sistema se utilizó el patrón CRUD (acrónimo de Crear, Obtener, Actualizar y Borrar del original en inglés: *Create*, *Read*, *Update and Delete*). El patrón CRUD Completo consiste en un caso de uso para administrar la información (CRUD Información). Definiéndose específicamente en los casos de uso: Gestionar Facultad, Gestionar Centro, Gestionar Laboratorio, Gestionar.

Se puso en práctica además el patrón de casos de uso actores múltiples, utilizado en el sistema para la autenticación de usuarios y la generación de reportes. Este patrón brinda la ventaja de representar a varios actores utilizando un mismo caso de uso con la creación de un nuevo actor a través de la generalización/ especialización.

Conclusiones

Después de identificado el proceso de negocio correspondiente al problema que se analiza, se modelaron los artefactos que genera la metodología OpenUp: modelo de negocio, el levantamiento de los requisitos funcionales y no funcionales, así como actores y casos de uso del sistema, descripciones de los mismos. Luego de haber quedado definido el diseño del sistema y las relaciones entre sus elementos, es posible dar comienzo el proceso de implementación de la “Sistema de control de acceso automatizado para los laboratorios de la UCI”.

Capítulo 3 Análisis y diseño de la aplicación

Introducción

El presente capitulo describe la arquitectura que se tendrá en cuenta para el diseño de la aplicación, haciendo énfasis en el estilo y patrón de arquitectura utilizados. Se modelan los artefactos que exige la Metodología OpenUp para esta fase, como por ejemplo, diagrama de clases, diagrama de interacción y diagrama de despliegue, además del modelo de base de datos; necesarios para lograr una mejor comprensión del funcionamiento de la aplicación que se decidió desarrollar.

3.1 Patrones de diseño y estilo de arquitectura

Los patrones y estilos son comportamientos que deben existir en el sistema, ayudan a describir qué es lo que debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. (39) Además los patrones de diseño expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software y expresan un esquema organizativo estructural fundamental para sistemas de software. (40)

3.1.1 Cliente-Servidor

En el mundo del Protocolo de Control de Transmisión y Protocolo de Internet (*Transmission Control Protocol* / *Internet Protocol*) TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama modelo Cliente-Servidor. Este término fue usado por primera vez en 1980 para referirse a las PC en red. Esta arquitectura se divide en dos partes claramente diferenciadas: servidor y clientes. Normalmente el servidor es una PC bastante potente que actúa de depósito de datos y funciona como un SGBD.

Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red. Esta tecnología proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso, en múltiples plataformas.

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). Es el proceso encargado de atender a múltiples clientes que hacen peticiones. Considerando la lógica de la aplicación que se desea desarrollar y analizando que el modelo cliente- servidor provee usabilidad, flexibilidad, interoperabilidad, escalabilidad en las comunicaciones, uso de entornos multiplataforma y recursos heterogéneos, se decidió utilizar este estilo arquitectónico para el desarrollo de la aplicación en su módulo de administración.

En el sistema propuesto se utiliza este patrón en la relación del agente con el servidor de agentes a través del protocolo tcp/ip y además con la relación de las estaciones de trabajo y el servidor apache mediante el protocolo http.

3.1.2 Modelo Vista Controlador (MVC)

Modelo Vista Controlador (Model - View - Controller) MVC, es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el SGBD y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

El sistema propuesto utiliza el framework CodeIgniter *(server-side)* y el framework ExtJs *(client-side)* que implementan internamente este patrón, separando la vista, del modelo y estos del controlador, para una mejor organización y reutilización del código.

3.1.3 Arquitectura N-Capas

Lo que se conoce como arquitectura en capas es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio, mecanismos de almacenamiento, presentando las siguientes ventajas:

- Desarrollos paralelos (en cada capa)
- Aplicaciones más robustas debido al encapsulamiento
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica)
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)
- Alta escalabilidad: La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.

Como tecnología, las arquitecturas de n-capas proporcionan una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios constantes. (41) En el sistema propuesto se pone de manifiesto la arquitectura N-capas con la integración de los patrones arquitectónicos descritos en los epígrafes anteriores.

3.2 Patrones de diseño (GRASP)

Los patrones de diseño son una manera práctica de describir ciertos aspectos de la organización de un programa, la conexión (relación) entre sus componentes, así como crea esquemas para definir estructuras que permiten construir sistemas de software. Durante el desarrollo de la aplicación utilizarán patrones básicos de asignación de responsabilidades como Patrones Generales de Software para Asignar Responsabilidades *(General Responsability Asignment Software Patterns)* ***GRASP.***

GRASP: El nombre se elige para indicar la importancia de captar (grasping) estos principios, si se quiere diseñar eficazmente el software orientado a objetos.

- **Experto:** Se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Creador:** Este patrón es el responsable de crear una nueva instancia de alguna clase. Asignarle a la clase B la responsabilidad de crear una instancia de clase A.
- **Alta Cohesión:** Asigna una responsabilidad de forma tal que la cohesión siga siendo alta.
- **Bajo Acoplamiento:** Este patrón es el encargado de asignar una responsabilidad para conservar bajo acoplamiento. Una clase con bajo acoplamiento no depende de muchas otras clases. Las clases con alto acoplamiento recurren a muchas clases y no es conveniente. Son más difíciles de mantener, entender y reutilizar.
- **Controlador:** Este patrón tiene la responsabilidad del manejo de mensajes de los eventos del sistema a una clase.

3.3 Diagramas de colaboración del análisis

Los diagramas de interacción del análisis son diagramas que describen como grupos de objetos colaboran para conseguir algún fin. En el modelo de análisis se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del Análisis. Las clases de análisis se centran en los requerimientos funcionales y se clasifican en Interfaz, de Control o Entidad.

3.3.1 Diagramas de colaboración de clases del análisis

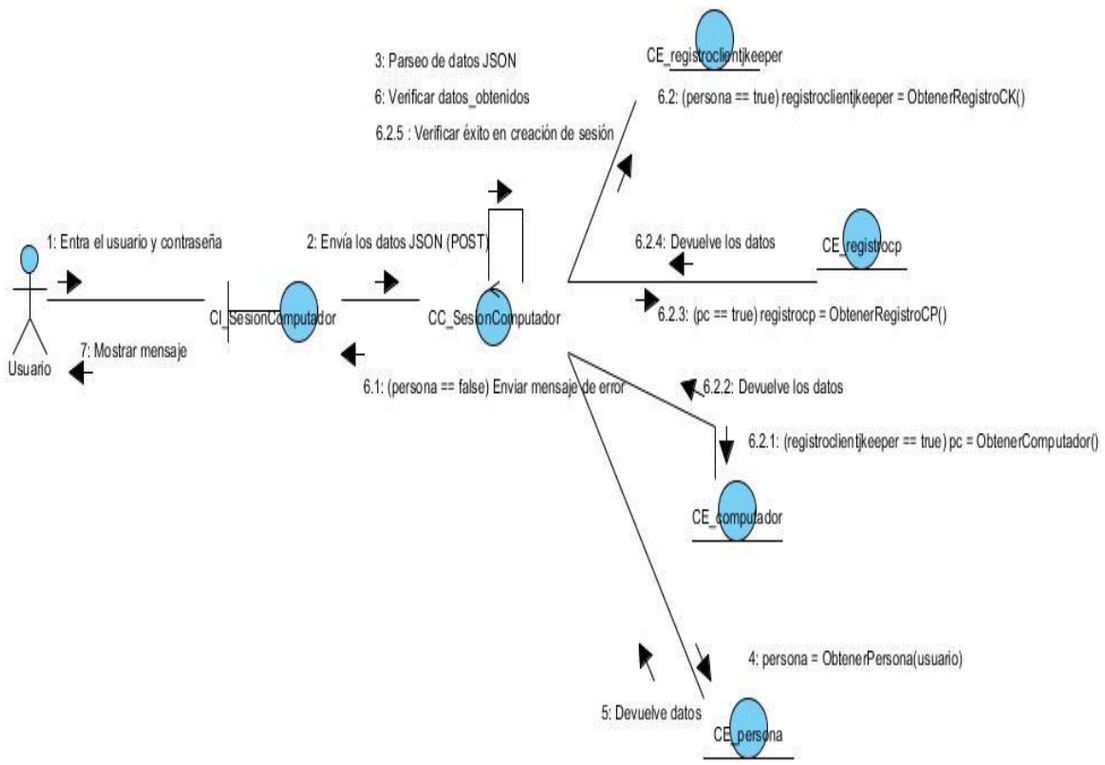


Figura 7 Diagrama de colaboración de clases del análisis para el caso de uso: "Crear sesión de computador".

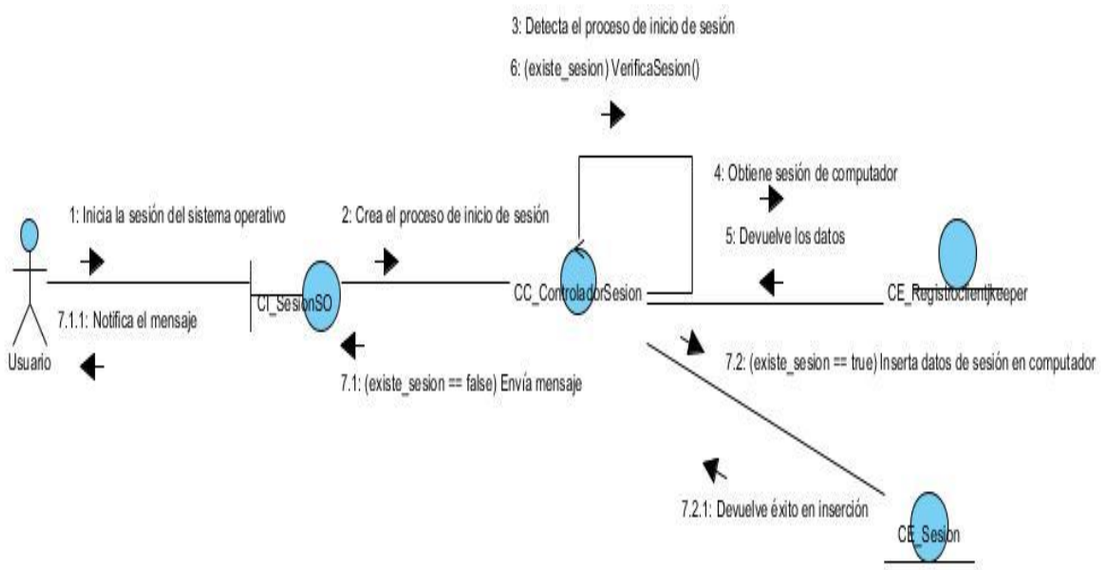


Figura 8 Diagrama de colaboración de clases del análisis para el caso de uso: "Verificar existencia de sesión de computador".

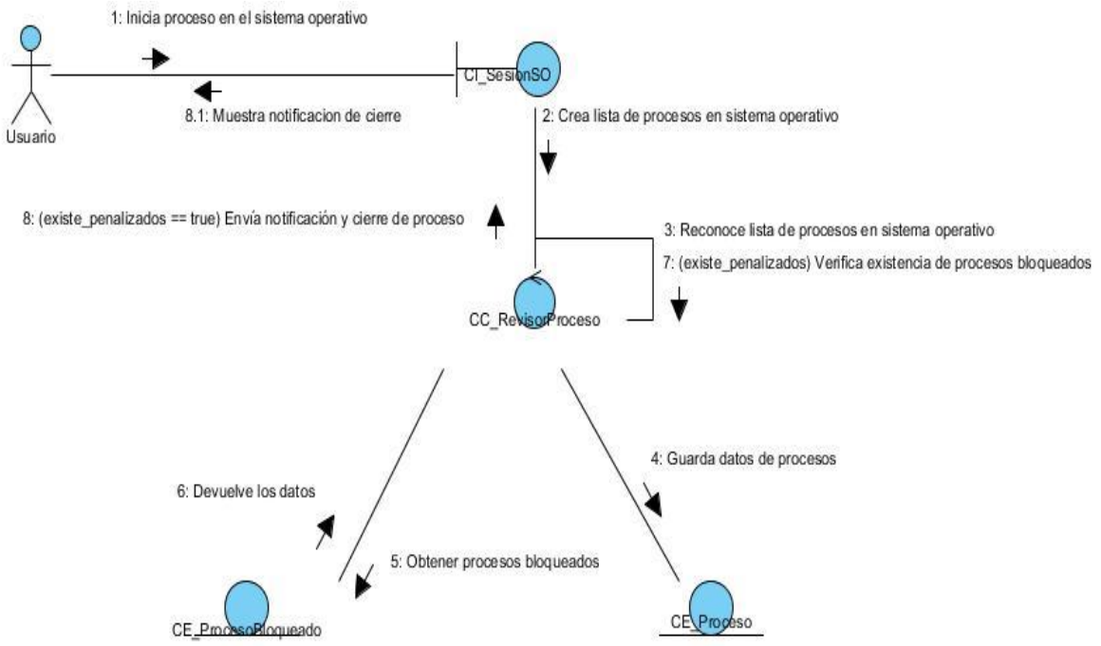


Figura 9 Diagrama de colaboración de clases del análisis para el caso de uso: "Cerrar proceso en el sistema operativo".

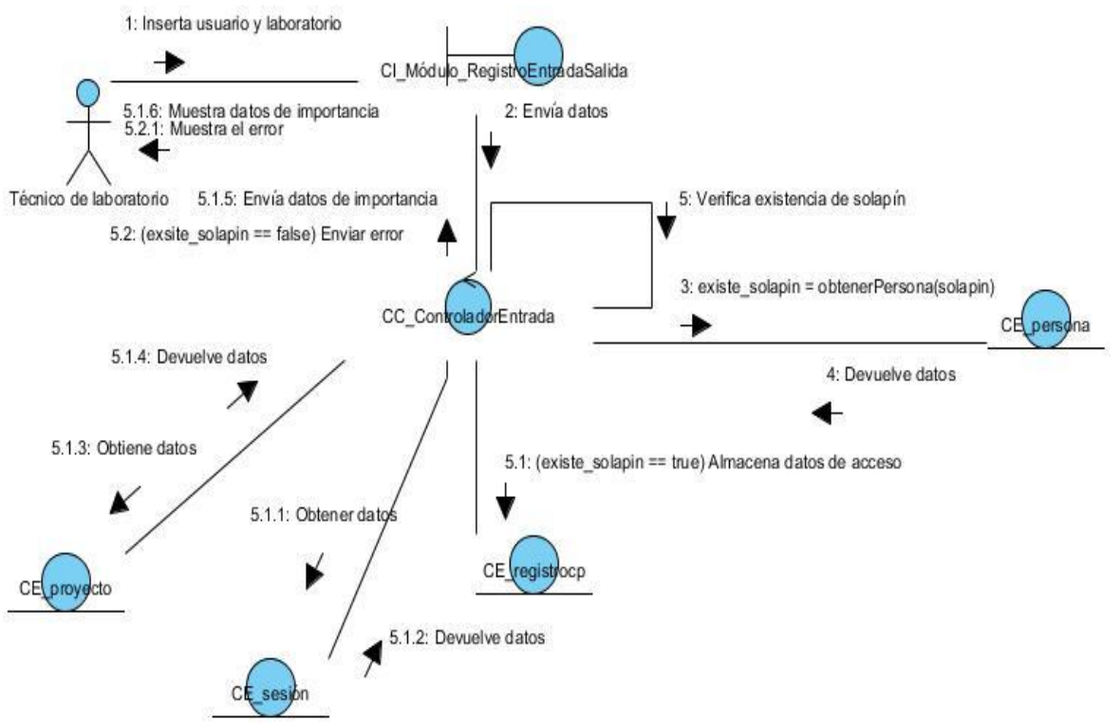


Figura 10 Diagrama de colaboración de clases del análisis para el caso de uso: "Controlar acceso de usuarios al laboratorio".

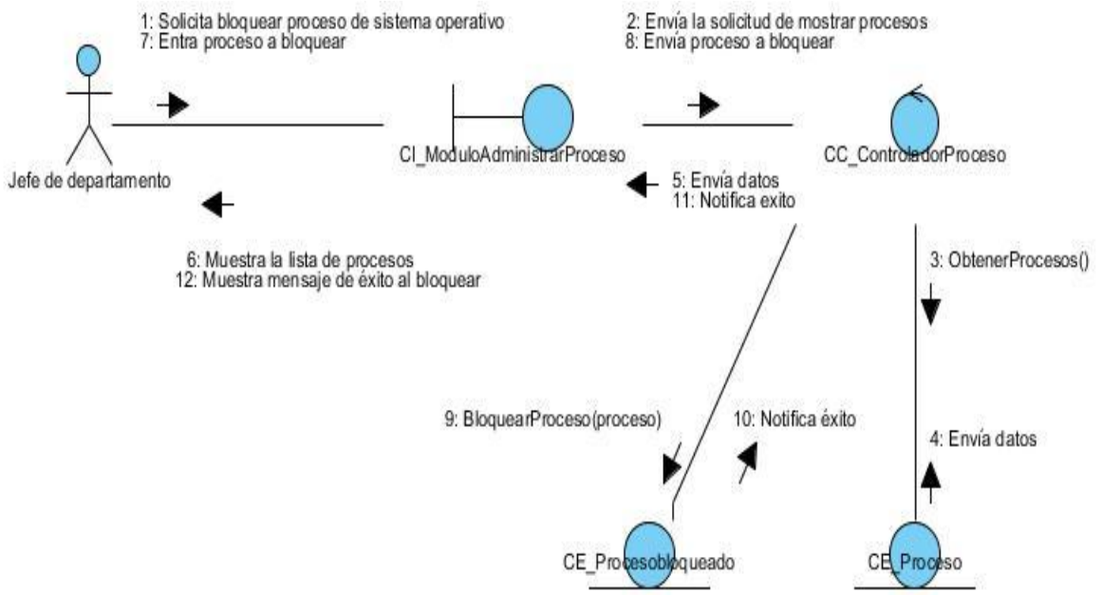


Figura 11 Diagrama de colaboración de clases del análisis para el caso de uso: "Bloquear proceso del sistema operativo".

3.4 Diagramas de clases del diseño

Un diagrama de clases del diseño es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. Describen gráficamente las especificaciones de las clases de software y las interfaces.

Diagrama de clases del diseño para los principales casos de uso

La figura 12 muestra el diagrama de clases del diseño correspondiente al caso de uso "Crear Sesión de computadores". En él se definen las clases que se utilizan para la autenticación del usuario, entre las que se encuentran la clase controladora AutenticarSesion, se modelan la clase Persona, Computador, RegistroCP, RegistroclientJkeeper como clases de acceso a datos, los formularios que visualizarán el cliente y las relaciones entre los elementos. Este caso de uso consume un servicio web de la UCI, este servicio web está disponible para todos los sistemas que se producen en la universidad.

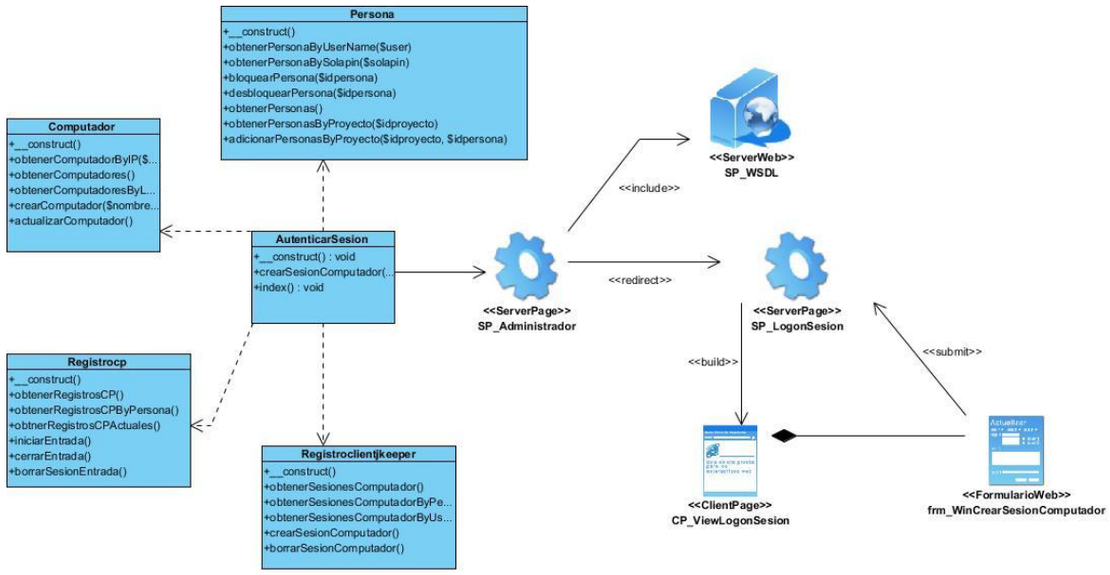


Figura 12 Diagrama de clases del diseño para el caso de uso "Crear sesión de computador"

La figura 13 representa el diagrama de clases del diseño correspondiente al caso de uso "Verificar existencia de sesión de computador", este diagrama representa clases del lenguaje java, que formaran parte de un agente que estará residente en las estaciones de trabajo, se muestra el uso de la librería SIGAR.JAR que permite la extracción y cierre de los procesos en los sistemas operativos. La clase ControladorSesion será un hilo de ejecución que verificará los parámetros de las sesiones en las estaciones de trabajo.

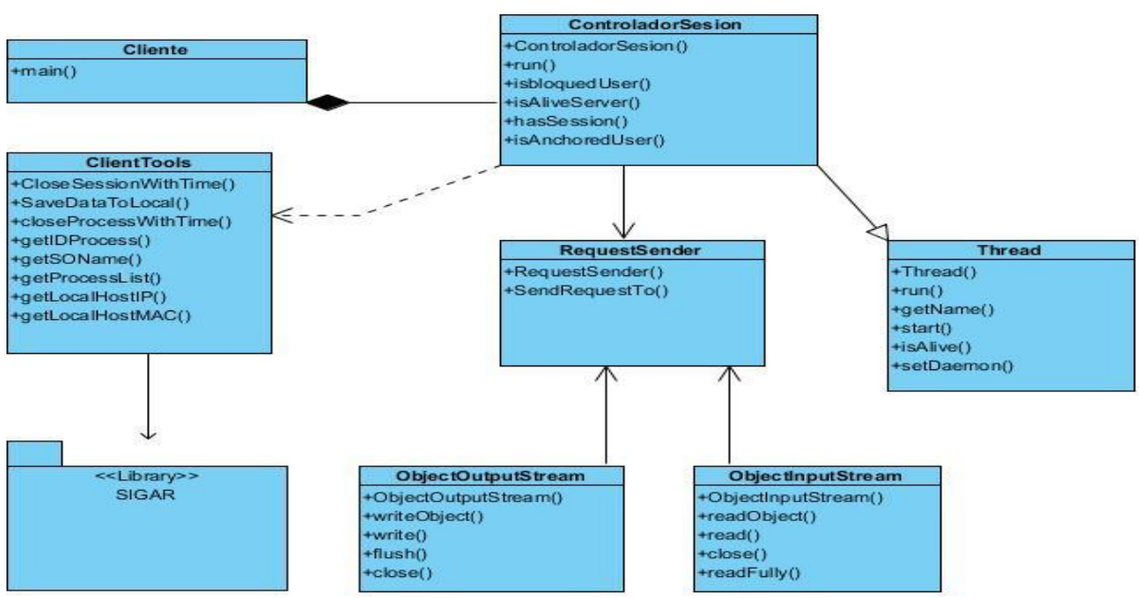


Figura 13 Diagrama de clases del diseño para el caso de uso "Verificar existencia de sesión de computador.

En la figura 14 se muestra una representación gráfica de la estructura del caso de uso "Cerrar proceso en el sistema operativo" el cual permite al sistema cerrar los procesos

penalizados de aplicaciones iniciadas por los usuarios. La clase RevisorProcesos es un hilo de ejecución que se mantiene revisando los procesos periódicamente y si existe algún proceso bloqueado, lo cierra y notifica al usuario.

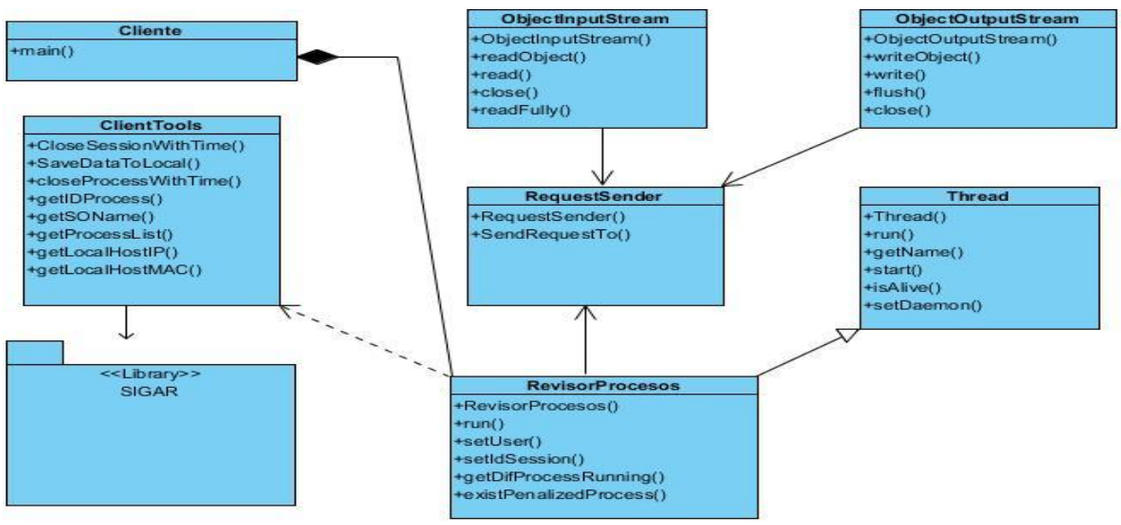


Figura 14 Diagrama de clases del diseño para el caso de uso "Cerrar proceso en el sistema operativo"

El caso de uso "Controlar acceso de usuarios al laboratorio" es representado en el diagrama de clases del diseño de la figura 16, las clases de acceso a datos son Persona, Registrocp, Sesión y Proyecto, las cuales son accedidas por la clase controladora CModuloEntrada.

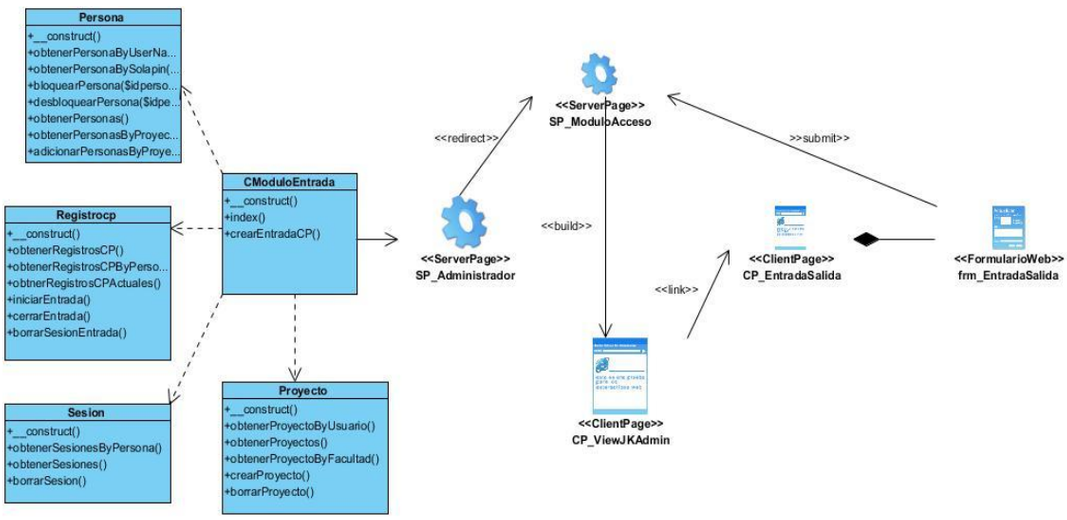


Figura 15 Diagrama de clases del diseño para el caso de uso "Controlar acceso de usuarios al laboratorio".

La figura 16 representa el diagrama de clases del diseño del caso de uso "Bloquear proceso del sistema operativo" donde la clase de acceso a datos es Proceso, la clase

controladora CModuloAdminProceso y las vistas CP_ViewLogonSesion que genera otra vista CP_AdminProcess.

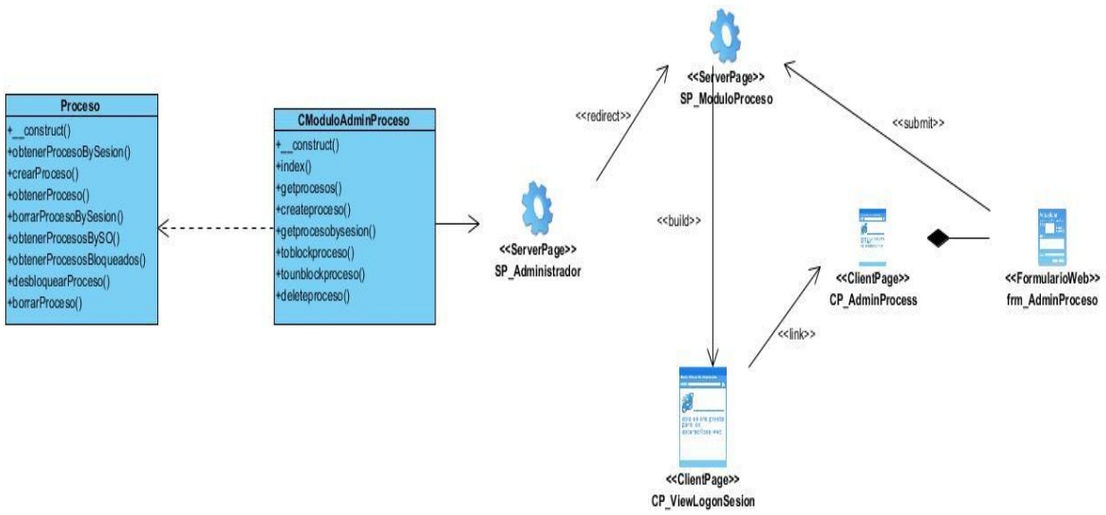


Figura 16 Diagrama de clases del diseño para el caso de uso "Bloquear proceso del sistema operativo".

3.5 Diseño de la base de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos correctamente diseñada permite obtener acceso a la información exacta y actualizada.

JKEEPER en su versión 1.0 posee una base de datos relacional encargada de hacer persistentes los datos necesarios para la generación de reportes y el control del uso de la tecnología por los usuarios. Para lograr el objetivo de este sistema se definieron 19 tablas. La base de datos permite el acceso desde dos puntos en el sistema, primero desde el servidor de agentes el cual guarda los reportes de los agentes, y segundo desde la aplicación de administración del sistema. (Ver anexo 2).

3.6 Diagrama de componentes

Los diagramas de componentes muestran los elementos de diseño de un sistema de software. Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos proporcionan y utilizan a través de las interfaces.

Describen los elementos físicos, muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos de código fuente, archivos, binarios, bibliotecas

cargadas dinámicamente o ejecutables. Las relaciones de dependencia se utilizan para indicar que un componente se refiere a los servicios ofrecidos por otro componente.

En el anexo 1 se representa una vista de la estructura de la implementación del sistema JKEEPER 1.0, con todos los paquetes, las relaciones de dependencias que se establecen entre los componentes, librerías utilizadas, así como cada una de las capas de la arquitectura definida. (Ver anexo 3).

Conclusiones

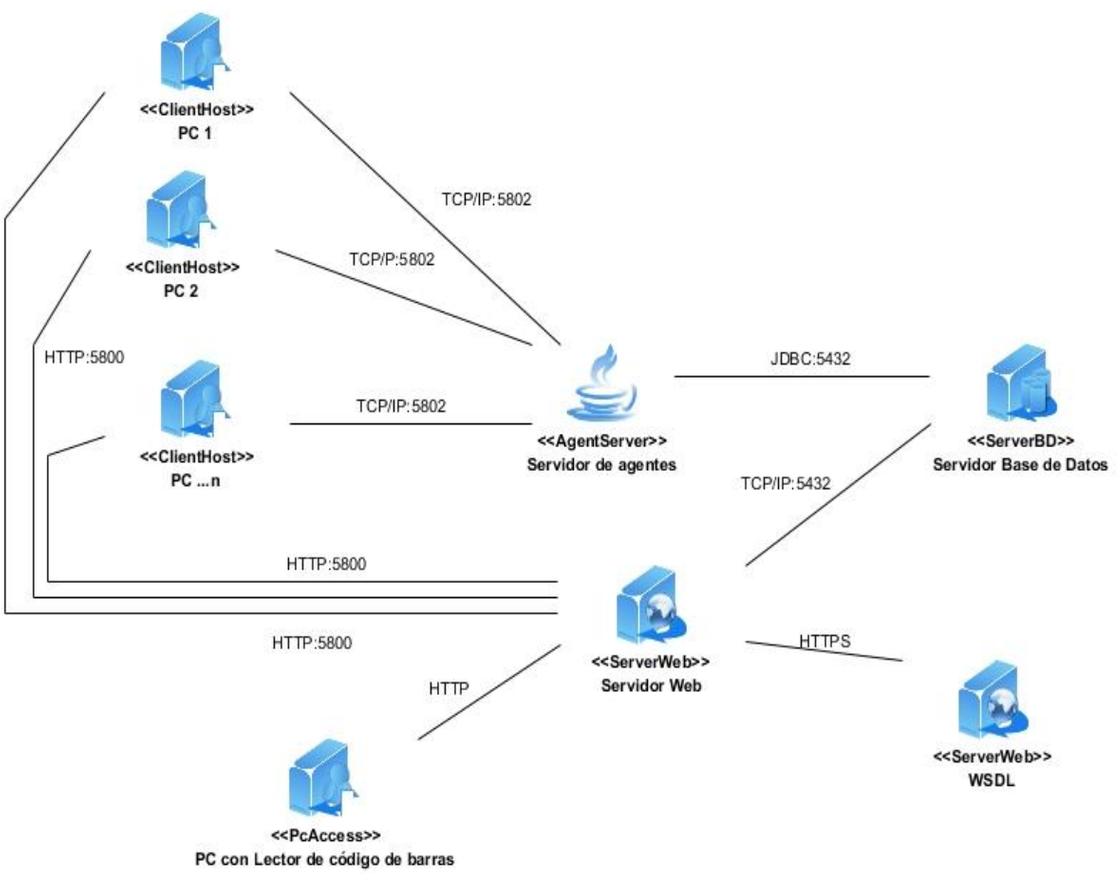
Al culminar este capítulo se logró alcanzar un entendimiento de los elementos a implementar y la estructura de estos a través de la realización de los diagramas de interacción y de clases del diseño. Se sentaron todas las bases para el éxito de la implementación del sistema propuesto.

Introducción

En el presente capítulo se realiza la implementación y el análisis de los resultados obtenidos durante el desarrollo de la aplicación. Se expone el diagrama de despliegue y la descripción de sus elementos, así como los casos de prueba o test de aceptación realizado al sistema, pues para lograr un producto con calidad es necesario implementar un plan de pruebas desde el principio, y así darle seguimiento a los cambios y desarrollar iterativamente. En este capítulo además de las pruebas se dan a conocer los resultados obtenidos hasta el momento.

4.1 Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos. Describen la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos, componentes de software. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.



4.1.1 Descripción de los nodos:

Figura 17 Diagrama de despliegue para el sistema JKEEPER 1.0.

- **Pc Cliente (*ClientHost*):** El nodo correspondiente a la computadora cliente representa todas aquellas PC que pueden ser utilizadas por los usuarios para acceder a la aplicación.
- **Servidor de Agentes (*AgentServer*):** El nodo Java representa al servidor de agentes, es el encargado de hacer la conexión a la base de datos y gestionar las peticiones de los agentes. Además de ser el responsable de controlar remotamente cada agente.
- **Servidor Web:** El nodo Servidor Web representa el servidor donde está alojada la aplicación web destinada al control administrativo y su correspondiente base de datos.
- **WSDL:** Este nodo representa gráficamente el servicio web que brinda la UCI a la comunidad universitaria.
- **Pc con lector de código de barras:** Este nodo corresponde a la estación de trabajo donde existirá una herramienta para la lectura del código de barra del solapín institucional, el acceso al laboratorio se controlará a través de esta computadora.
- **Servidor de Base de datos:** Este nodo representa la computadora donde se existirá el servidor de base de datos.

4.1.2 Descripción del tipo de comunicación

La comunicación entre los nodos que representan las Pc Clientes y el Servidor de Agentes (*AgentServer*) se realiza mediante el protocolo TCP/IP. Se le denomina conjunto de protocolos **TCP/IP**, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP). El conjunto TCP/IP está diseñado para enrutar y tiene un grado muy elevado de fiabilidad, es adecuado para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores web. Es compatible con las herramientas estándar para analizar el funcionamiento de la red.

Existe además la comunicación entre los nodos que representan las Pc Clientes y el Servidor Web, este se realiza mediante el protocolo HTTP. Protocolo a nivel de aplicación con la agilidad y velocidad necesaria para sistemas de información distribuidos, colaborativos y de hipermedia. Es un protocolo orientado a objetos, genérico, que puede usarse para muchas tareas extendiendo sus métodos. Una característica de HTTP es que permite que los sistemas se construyan independientemente de la información que se transfiere.

La comunicación que establecen el Servidor de Agentes (*AgentServer*) y la base de datos es a través de la API JBDC, una interfaz de acceso a RDBMS (*Relational Database Management System*) independiente de la plataforma y del gestor de bases de datos utilizado. Esta librería posee diferentes manejadores de conexiones dedicados a un

modelo de base de datos en particular, este son conjuntos de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos que pueden manejar (42).

Los nodos correspondientes al Servidor Web y WSDL utilizan el protocolo de comunicación seguro HTTPS, como protocolo de seguridad dado el alto grado de confidencialidad que requieren las contraseñas de los usuarios que acceden a la aplicación. HTTPS utiliza un cifrado basado en la Capa de Conector Seguro (*Secure Socket Layer*) y *Capa de Transporte Seguro (Transport Layer Security)* SSL/TLS para crear un canal cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente.

El nodo Servidor Web estable comunicación con el Servidor de BD mediante el protocolo de Acceso a Objetos de Datos (*ActiveX Data Objects*) ADO. Protocolo basado en el modelo de objetos, el cual define una jerarquía de objetos programables que pueden ser usados por desarrolladores de páginas Web para acceder a la información almacenada en una base de datos.

4.2 Pruebas de aceptación

Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al cliente. La utilización de estas, proporcionan grandes ventajas, permitiendo a los programadores principalmente medir la calidad de su trabajo y garantizar la entrega de un producto con calidad y en correspondencia con las necesidades del cliente. Se definieron casos de prueba para todos los casos de uso, a continuación se dan a conocer las pruebas que se realizaron a los casos de uso de alta prioridad con las que cuenta el sistema JKEEPER.

1. Caso de prueba de aceptación para el CU: “Autenticar usuario”

Caso de prueba de aceptación	
Código Caso de Prueba: JKEEPER-01-1	**Nombre del CU**: Autenticar usuario.
Descripción de la Prueba: Esta prueba consiste en realizar la autenticación del usuario en el sistema web de administración JKEEPER.	
Condiciones de Ejecución: Para que esto sea posible el usuario primero abrir el navegador y entrar la URL de la aplicación.	
Entrada / Pasos de ejecución:	

Entrada: El usuario entrará el nombre de usuario y la contraseña EC1.1 Autenticar satisfactoriamente: Para que el usuario pueda autenticarse debe introducir en los campos de texto el nombre de usuario y la contraseña, después de llenados los campos debe presionar Aceptar. EC1.2 Autenticar insatisfactoriamente: El usuario introduce en los campos de texto el nombre de usuario y la contraseña incorrectamente para el sistema, después de llenados los campos debe presionar Aceptar. EC1.3 Campos vacíos: El usuario deja en blanco los campos de texto de nombre de usuario y/o contraseña, luego debe presionar Aceptar.
El sistema verifica los datos, si son válidos ver resultado esperado para EC1.1, si no son válidos ver resultado esperado para EC1.2 y si existen campos nulos ver resultado para EC1.3. **Resultado Esperado:** EC1.1: El sistema otorga los permisos de acceso al usuario y muestra la vista de administración. EC1.2: El sistema muestra una notificación de error “El usuario no existe en la base de datos del sistema, contacte al administrador”. (ver anexo 4) EC1.3: El sistema marca de rojo los campos nombre de usuario y contraseña señalando que los campos son obligatorios.
Evaluación de la Prueba: Satisfactoria

2. Caso de prueba de aceptación para el CU: ”Bloquear proceso”

Caso de prueba de aceptación	
Código Caso de Prueba: JKEEPER-02-1	**Nombre del CU**: Bloquear proceso.
Descripción de la Prueba: Esta prueba consiste en bloquear un proceso del sistema operativo en las estaciones de trabajo, mediante la aplicación de administración del sistema JKEEPER.	
Condiciones de Ejecución: Para que esto sea posible el usuario autenticarse en la aplicación de administración JKEEPER, y acceder al módulo Proceso.	
Entrada / Pasos de ejecución:	

Entrada: El usuario selecciona un proceso en la lista de procesos. EC1.1 Bloquear satisfactoriamente: el usuario debe presionar Bloquear proceso y debe entrar una descripción (opcional) de los motivos del bloqueado.
Resultado Esperado: EC1.1: El sistema marca de rojo el proceso bloqueado, mostrando la notificación “Los procesos seleccionados han sido bloqueados”, cambia el botón para el estado de Desbloquear proceso. (ver anexo 5)
Evaluación de la Prueba: Satisfactoria

3. Caso de prueba de aceptación para el CU: “Anclar persona”

Caso de prueba de aceptación	
Código Caso de Prueba: JKEEPER-03-1	**Nombre del CU**: Anclar persona.
Descripción de la Prueba: Esta prueba consiste en anclar un usuario a una estación de trabajo mediante su dirección física MAC.	
Condiciones de Ejecución: Para que esto sea posible el usuario debe autenticarse en la aplicación de administración JKEEPER, y acceder al módulo Persona.	
Entrada / Pasos de ejecución: **Entrada:** El usuario selecciona una persona en la lista de personas. EC1.1 Anclar satisfactoriamente: el usuario debe presionar Anclar persona, y entrar el código MAC, después debe presionar Aceptar. EC1.2 Campos vacíos: El usuario deja en blanco el campo de texto nombre de MAC, luego debe presionar Aceptar.	
El sistema verifica los datos, si son válidos ver resultado esperado para EC1.1 y si existen campos nulos ver resultado para EC1.2. **Resultado Esperado:** EC1.1: El sistema ancla la persona seleccionada al código MAC, notifica al usuario con el mensaje “La(s) persona(s) seleccionada(s) han sido ancladas”, el botón cambia de estado a Desanclar persona. EC1.2: El sistema muestra una notificación de error “Usted debe entrar algún código	

MAC”.(ver anexo 6)
Evaluación de la Prueba: Satisfactoria

4. Caso de prueba de aceptación para el CU: “Controlar acceso de usuarios al laboratorio”.

Caso de prueba de aceptación	
Código Caso de Prueba: JKEEPER-04-1	**Nombre del CU**: Controlar acceso de usuarios al laboratorio.
Descripción de la Prueba: Esta prueba consiste en proporcionar el acceso a los usuarios a un laboratorio específico.	
Condiciones de Ejecución: Para que esto sea posible el usuario debe autenticarse en la aplicación de administración JKEEPER, y acceder al módulo Controlador de Entradas.	
Entrada / Pasos de ejecución: **Entrada:** El usuario entra un solapín y selecciona el laboratorio requerido para el acceso. EC1.1 Proporcionar acceso satisfactoriamente: después de llenar los campos de solapín y laboratorio correctamente, el usuario debe presionar aceptar. EC1.2 Proporcionar acceso insatisfactoriamente: después de llenar los campos de solapín y laboratorio incorrectamente, el usuario debe presionar aceptar. EC1.3 Campos vacíos: El usuario deja en blanco el campo de texto solapín y el campo de texto laboratorio, debe presionar aceptar.	
El sistema verifica los datos, si son válidos ver resultado esperado para EC1.1, si no son válidos ver resultado esperado para EC1.2 y si existen campos nulos ver resultado para EC1.3. **Resultado Esperado:** EC1.1: El sistema concede los permisos de acceso al usuario, actualiza los campos de la vista, mostrando la foto y datos de interés. EC1.2: El sistema muestra una notificación de error “El usuario con ese solapín no pertenece al laboratorio indicado, contacte al administrador”. EC1.3: El sistema muestra una notificación de error “Usted debe entrar los datos correspondientes: solapín y laboratorio”. (ver anexo 7)	

Evaluación de la Prueba: Satisfactoria

5. Caso de prueba de aceptación para el CU: “Crear sesión de computador”.

Caso de prueba de aceptación	
Código Caso de Prueba: JKEEPER-05-1	**Nombre del CU**: Crear sesión de computador.
Descripción de la Prueba: Esta prueba consiste en crear una sesión a través de las estaciones de trabajo, accediendo a la URL y creando una sesión de computador, posibilitando así, que el agente reconozca al usuario.	
Condiciones de Ejecución: Para que esto sea posible el usuario debe acceder a la URL del sistema JKEEPER, donde podrá crear la sesión de computador.	
Entrada / Pasos de ejecución: **Entrada:** El usuario entrará el nombre de usuario y la contraseña. .EC1.1 Proporcionar acceso satisfactoriamente: después de llenar los campos de usuario y contraseña correctamente, el usuario debe presionar aceptar. EC1.2 Proporcionar acceso insatisfactoriamente: después de llenar los campos de usuario y contraseña incorrectamente, el usuario debe presionar aceptar. EC1.3 Campos vacíos: El usuario deja en blanco el campo de texto usuario y/o el campo de texto contraseña, debe presionar aceptar.	
El sistema verifica los datos, si son válidos ver resultado esperado para EC1.1, si no son válidos ver resultado esperado para EC1.2 y si existen campos nulos ver resultado para EC1.3. **Resultado Esperado:** EC1.1: El sistema crea una nueva sesión de computador, notificando al usuario lo siguiente: “Se ha creado una sesión para el cliente JKEEPER en el IP xxxxx con el usuario xxxxxx”. EC1.2: El sistema muestra una notificación de error según sea el caso: • Si existe una sesión para esa estación de trabajo, notifica un error y muestra el mensaje “Ya existe una sesión para el cliente JKEEPER en el IP xxxxx y el usuario xxxxxx”. • Si el usuario no se haya registrado en el Controlador de entradas, notifica un error y muestra el mensaje “Usted debe registrarse en el controlador de entrada para crear	

sesiones en las estaciones de trabajo, contacte al administrador".

- Si el usuario no este registrado en el sistema, notifica un error y muestra el mensaje "El usuario no existe en el sistema, contacte al administrador".
- Si la estación de trabajo no exista en el sistema, notifica un error y muestra el mensaje "El computador desde el que hace referencia no existe, contacte al administrador, IP: xxxxx". (ver anexo 8)

EC1.3: El sistema muestra una notificación de error "Usted debe llenar los campos correctamente".

Evaluación de la Prueba: Satisfactoria

Se realizaron además pruebas de estrés y pruebas de inyección SQL, las primeras con el objetivo de obtener datos, sobre la carga del sistema, que ayuden a realizar el dimensionamiento del sistema, pues esta prueba genera carga en el sistema hasta hacerlo inutilizable. La segunda, realizada con el objetivo de la inserción o la "inyección" de una consulta SQL a través de los datos de entrada de la aplicación del cliente.

El éxito de explotar una inyección SQL puede ser el de leer datos sensibles de la base de datos, modificar la base de datos para (Insertar / Actualizar / Borrar), ejecutar operaciones de administración de la base de datos (por ejemplo parar el DBMS), recuperar el contenido de un archivo presente en el sistema de ficheros y/o DBMS, Y hasta en algunos casos, llegar a la shell (línea) de comandos del sistema operativo.

Una vez culminada la ejecución de las pruebas de aceptación el cliente quedo satisfecho en un 100 por ciento con respecto a las funcionalidades que ofrece JKEEPER en su versión 1.0. De la culminación del proceso de estas pruebas se emitió una carta de aceptación por el cliente avalando lo anterior descrito. (Ver anexo 9)

4.3 Pruebas funcionales

Para emitir la liberación del sistema se realizaron pruebas funcionales por el Grupo de Calidad de CEIGE. Se ejecutaron las pruebas con 5 probadores con el administrador de calidad al frente en 5 ordenadores, cada computadora con el agente instalado así como acceso a la administración del sistema JKEEPER en la web.

Una vez finalizado el proceso de pruebas se arribó a las siguientes conclusiones:
Fueron realizadas tres iteraciones arrojando para una primera iteración de un total 28 No Conformidades, 25 de tipo Validación y 3 No Conformidades del tipo Opciones que no funcionan, en una segunda iteración fueron encontradas 6 No Conformidades, 4 No

Conformidades del tipo Validación y 2 no conformidad del tipo Opciones que no funcionan mientras que para una tercera iteración no se encontraron No Conformidades. El sistema fue liberado satisfactoriamente con un acta de liberación. A continuación se muestra una imagen que aprueba lo anteriormente descrito:

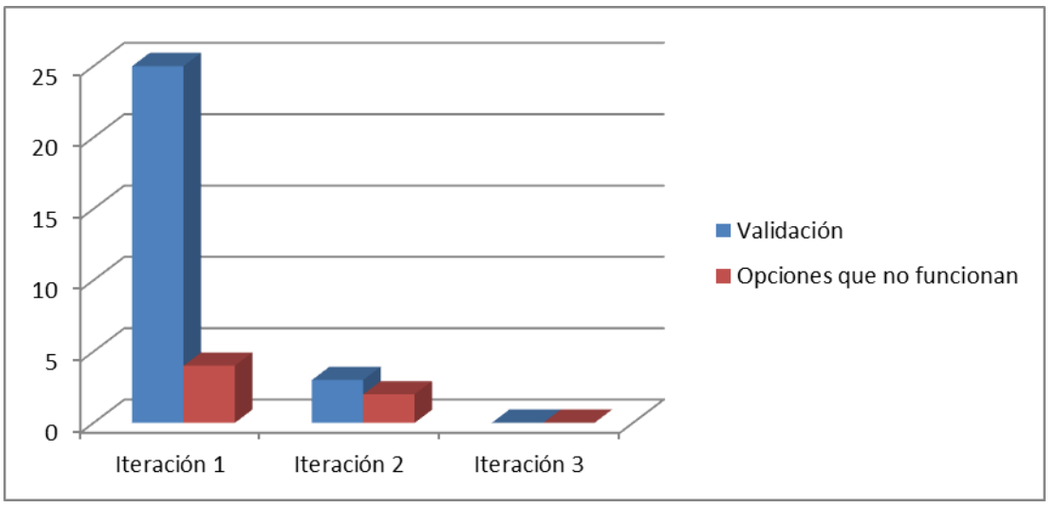


Figura 18 Gráfico de no conformidades detectadas

El Grupo de Calidad emitió un aval donde se exponen que el software esta Liberado (ver anexo 10). Culminada la liberación la aplicación se desplegó el 18 de febrero del 2013 en el Centro de Telemática de la facultad 2, para el montaje del servidor de agentes, la base de datos y el servidor apache se utilizaron máquinas virtuales, mientras que los agentes se desplegaron en computadoras ordinarias.

4.4 Aporte social y económico

El Sistema JKEEPER 1.0 para los laboratorios en la UCI no es un software con fines comerciales, pues está orientado a resolver los problemas existentes en la Universidad, aunque puede ampliarse para convertirlo en una solución general, capaz de aplicarse a cualquier empresa o institución. Su principal objetivo es controlar el control de acceso de las personas a los laboratorios, por tanto, los beneficios inmediatos son mayormente intangibles:

- Ahorro de tiempo en la gestión de acceso al laboratorio.
- Control estricto sobre las estaciones de trabajo y sus aplicaciones.
- Aumento de la organización interna.

De forma directa también se obtendrán beneficios tangibles ya que una vez desplegado el sistema se podrán controlar con mayor facilidad los accesos de las personas a los laboratorios, siendo así posible una mejor planificación de la cantidad de estaciones de trabajo en prestación de servicio, lo que propicia un alto rendimiento en la productividad.

Además con la implementación del Sistema JKEEPER 1.0 se sustituye el uso de papel en gran parte de los procesos que se desarrollan en el control de acceso a los laboratorios actualmente contribuyendo así, al proyecto que lleva la universidad para el ahorro de recursos. Además, facilita y organiza el trabajo en un menor espacio de tiempo y posibilita información más confiable y disponible.

Conclusiones

Al concluir el presente capítulo se logró completar la fase de implementación, se concretaron los diagramas de despliegue y se mostraron algunas imágenes o pantallas de la aplicación durante su funcionamiento. Al concluir el desarrollo de las pruebas al sistema se demuestra con el análisis de los resultados obtenidos, que las funcionalidades alcanzadas por JKEEPER 1.0, se han desarrollo de acuerdo a los requerimientos definidos en la etapa inicial y en el período establecido. Funcionalidades que reflejan información de gran importancia para el control de acceso a los laboratorios de la UCI.

Conclusiones generales

- Los controladores de acceso utilizados actualmente a nivel mundial y nacional no satisfacen las necesidades de la UCI.
- Con la realización del análisis y el diseño de la aplicación se logró un acercamiento de la misma para su posterior implementación.
- JKEEPER en su versión 1.0 controla las estaciones de trabajo y el acceso a los laboratorios alcanzando la supervisión sobre las aplicaciones ejecutadas en cada estación de trabajo.
- Los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente ya que la aplicación da solución a la situación problémica que le dio origen.

Recomendaciones

- Realizar el despliegue del sistema propuesto a todas las áreas de laboratorios de la Universidad de las Ciencias Informáticas.
- Desarrollar la segunda versión de JKEEPER, incluyéndole nuevas funcionalidades.

Referencias bibliográficas

1. Katz,R. *El papel de las tecnologias de la informacion y las comunicaciones en el desarrollo economico social.* Madrid : Ariel, S.A, 2009. 978-841-11-43.

2. Carlisle Adams, Steve Lloyd. *Understanding PKI: Concepts, Standards and Deployment Considerations.* Boston : Pearson Education, 2003. 0-672-32391-5.

3. Clercq, J De. *Single Sing-On Architectures*. s.l. : Berlin Heidelberg, 2002, págs. 40,58.

4. A.J. Stell, Dr R.O. Sinnott, Dr J.P. Watt**.** *Comparison of Advanced Authorisation Infrastructures for Grid Computing.* s.l. : 1ra ed. IEEE, 2005. ISBN 1550-5243.

5. Araujo Brett A, Bravo V**.** *Autenticación, Control de Acceso, Autorización.* s.l. : CENTIDEL, 2008.

6. Areitio Bertolín, J. *Seguridad de la Información: Redes, informática y sistemas de información*. s.l. : Madrid. ParaInfo. 2008. ISBN 8497325028

7. knowledgrEs. *Control de Acceso basado en el papel*. [En línea] [Citado el: 15 de Noviembre de 2012.] http://www18.knowledgres.com/00044992/ControlDeAccesoBasadoEnElPapel.

8. ECURED**.** *Control de Acceso* - EcuRed. [En línea] [Citado el: 06 de 11 de 2012.] http://www.ecured.cu/index.php/Control_de_acceso.

9. GINSATEC S.A de C.V**.** *Control de Acceso y Personal*. [En línea] GINSATEC. [Citado el: 8 de 11 de 2012.] http://www.ginsatec.com.mx/index.php?option=com_content&view=article&id=9&Itemid=23&lang=es.

10. WilCox. *Ingenieria Electronica*. [En línea] [Citado el: 21 de Noviembre de 2012.] http://www.wilcox.com.ar/productos.php?q=ctrlacceso.

11. Cosentino, L. *Control de accesos, conceptos, historia y esquema básico.* 74, Argentina : RNDS, 2013. [Citado el: 8 de noviembre del 2012]. http://www.rnds.com.ar/articulos/045/RNDS_152W.pdf

12. RedHat. *Portal de clientes*. [En línea] [Citado el: 10 de noviembre de 2012.] https://access.redhat.com/knowledge/docs/es-ES/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/selg-overview.html.

13. RantRing. *Soluciones para sistemas de seguridad*. [En línea] [Citado el: 20 de noviembre de 2012.] http://rantring.com/seg_tarjP.htm.

14. Gerber, M. *Information Management & Computer Security*. [En línea] Emerald. [Citado el: 12 de 10 de 2012.] http://www.emeraldinsight.com/journals.htm?articleid=862784&show=html&WT_mc_id=alsoread&PHPSESSID=iflfnelq87q2bi1rm3lgs2moa2&&nolog=137421.

15. Gutiérrez, C. *A Survey of Web Services Security*. [En línea] 2004. [Citado el: 09 de 10 de 2012.] http://mmlabold.ceid.upatras.gr/courses/AIS_SITE/files/3%5CA%20Survey%20of%20Web%20Services%20Security.pdf. 978-3-540-22054-1.

16. Lucena López, Manuel J. *UNED*. [En línea] Mayo de 2003. [Citado el: 1 de 11 de 2012.] http://www.uned.es/413042/material/Criptografia.pdf.

17. Menezes, Alfred J. *Handbook of applied cryptography*. [En línea] Octubre de 1996. [Citado el: 05 de 11 de 2012.] http://cacr.uwaterloo.ca/hac/.

18. Montenegro, L. *Seguridad de la Información: Más que una actitud, un estilo de vida.* [En línea] MVP. [Citado el: 12 de 10 de 2012.] http://www.microsoft.com/conosur/technet/articulos/seguridadinfo/.

19. Kimaldi. *Controlador de Acceso*. [En línea] [Citado el: 15 de noviembre de 2012.] http://www.kimaldi.com/empresa/quienes_somos.

20. CNX ANISTER . *Distribuidor global de soluciones de seguridad.* [En línea] [Citado el: 21 de noviembre de 2012.] http://www.anixtersoluciones.com/latam/cl/seguridad/18043/migre_hacia_la_mayor_tecnologia_en_control_de_accesos_es.htm.

21. AMAG TECHNOLOGY. *Javelin form AMAG technology*. [En línea] [Citado el: 20 de noviembre de 2012.] http://www.securityinfowatch.com/product/10327856/amag-technology-javelin.

22. SmartCard Systems S.A. *Sistemas de control de acceso y software de control de acceso, HID.* [En línea] 2 de Enero de 2013. [Citado el: 15/12/ 2012.] http://www.scssa.com.ar/control-de-acceso.htm.

23. Arquero Sistema corporativo. *Software de control y seguridad. Control de acceso y control de horario.* [En línea] 15 de Enero de 2013. [Citado el: 16/ 12/2012.] http://www.sci-spain.com/root.php?modulo=controlAcceso.

24. Binary Ingeniería y Software S.L. *Binary Ingeniería y Software Sistemas de control de acceso.* [En línea] 14 de Febrero de 2013. [Citado el: 18 de Diciembre de 2012.] http://www.binaryis.com/ControlAccesos.aspx.

25. GSInnova. *Rational Rose Enterprise*. [En línea] [Citado el: 1 de diciembre de 2012.] http://www.rational.com.ar/herramientas/roseenterprise.html.

26. Visual Paradigm. *UML, BPMN y Database Tools for Software Development.* [En línea] [Citado el: 2 de diciembre de 2012.] http://www.visual-paradigm.com/.

27. eclipse. *About the eclipse fundation*. [En línea] [Citado el: 23 de noviembre de 2012.] http://www.eclipse.org/org/.

28. Clayberg E, Rubel D. *Eclipse: Building Commercial-Quality Plug-ins, Second Edition.* Julio 2007. ISBN-10: 0321553462.

29. NetBeans. *NetBeans IDE*. [En línea] [Citado el: 23 de noviembre de 2012.] http://netbeans.org/community/releases/72/.

30. MySQL. *MySQL The world's most popular open source database.* [En línea] [Citado el: 13 de noviembre de 2012.] http://dev.mysql.com/doc/index-about.html.

31. EllisLab. *EllisLab CodeIgniter*. [En línea] [Citado el: 12 de diciembre de 2012.] http://ellislab.com/codeigniter.

32. Henst, C Van Der. *Maestros del Web*. [En línea] [Citado el: 14 de diciembre de 2012.] http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/.

33. Larman, C. *Applying Uml and Patterns: An Introduction to Object-Oriented Analysis and Design, and the Unified Process.* s.l. : Prentice Hall Professional, 2002. ISBN 0-13-092569-1.

34. Aguilar, C. *Aplicación de conceptos de gestión de proyectos y gestión de riesgos en el desarrollo de productos nuevos en el campo de la tecnología de la información.* Mayaguez : Universidad de Puerto Rico, Diciembre 2005.

35. E. Jeffries R, Anderson A, Hendrickson C. *Extreme Programing Installed.* s.l. : Addison-Wesley Professional, 2001. ISBN 0-201-70842-6.

36. EPF. *OpenUp Basic Eclipse Foundation*. [En línea] 30 de 11 de 2011. http://epf.eclipse.org/wikis/openupsp/.

37. GSInnva. *Rational Unified Process*. [En línea] [Citado el: 7 de Diciembre de 2012.] http://www.rational.com.ar/herramientas/rup.html.

38. Jacobson I, Booch G, Rumbaugh J. *El proceso unificado de desarrollo de software.* La Habana : Felix Valera , 2004. págs. 128,129. Vol. 1. 0-201-54435-0.

39. Tedeschi, N. MSDN. [En línea] Microsoft, 26 de 5 de 2009. [Citado el: 1 de 12 de 2012.] http://msdn.microsoft.com/es-es/library/bb972240.aspx.

40. EcuRed. *Patrones de Diseño y Arquitectura*. [En línea] EcuRed, 3 de 12 de 2012. [Citado el: 4 de 12 de 2012.] http://www.ecured.cu/index.php/Patrones_de_dise%C3%B1o_y_arquitectura.

41. Santiago Domingo Moquillaza H, Vega Huerta H. *Revista de Investigación de Sistemas e Informática* .Universidad Nacional Mayor de San Marcos, 2010. 1816-3823.

42. Thomas, T. M. Java Data Access. en: Java *Data Access JDBC, JNDI, and JAXP*. MINDS, H. New York, M&T Books An imprint of Hungry Minds, Inc., 2002. 1: 2.p. ISBN 0-7645-4864-8.

Bibliografía

1. Araujo Brett A, Bravo V**.** *Autenticación, Control de Acceso, Autorización.* s.l. : CENTIDEL, 2008.

2. Areitio Bertolín, J. *Seguridad de la Información: Redes, informática y sistemas de información*. s.l. : Madrid. ParaInfo. 2008. ISBN 8497325028

3. knowledgrEs. *Control de Acceso basado en el papel*. [En línea] [Citado el: 15 de Noviembre de 2012.] http://www18.knowledgres.com/00044992/ControlDeAccesoBasadoEnElPapel.

4. ECURED**.** *Control de Acceso* - EcuRed. [En línea] [Citado el: 06 de 11 de 2012.] http://www.ecured.cu/index.php/Control_de_acceso.

5. GINSATEC S.A de C.V**.** *Control de Acceso y Personal*. [En línea] GINSATEC. [Citado el: 8 de 11 de 2012.] http://www.ginsatec.com.mx/index.php?option=com_content&view=article&id=9&Itemid=23&lang=es.

6. WilCox. *Ingenieria Electronica*. [En línea] [Citado el: 21 de Noviembre de 2012.] http://www.wilcox.com.ar/productos.php?q=ctrlacceso.

7. Cosentino, L. *Control de accesos, conceptos, historia y esquema básico.* 74, Argentina : RNDS, 2013. [Citado el: 8 de noviembre del 2012]. http://www.rnds.com.ar/articulos/045/RNDS_152W.pdf

8. RedHat. *Portal de clientes*. [En línea] [Citado el: 10 de noviembre de 2012.] https://access.redhat.com/knowledge/docs/es-ES/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/selg-overview.html.

9. RantRing. *Soluciones para sistemas de seguridad*. [En línea] [Citado el: 20 de noviembre de 2012.] http://rantring.com/seg_tarjP.htm.

10. Gerber, M. *Information Management & Computer Security*. [En línea] Emerald. [Citado el: 12 de 10 de 2012.] http://www.emeraldinsight.com/journals.htm?articleid=862784&show=html&WT_mc_id=alsoread&PHPSESSID=iflfnelq87q2bi1rm3lgs2moa2&&nolog=137421.

11. Gutiérrez, C. *A Survey of Web Services Security*. [En línea] 2004. [Citado el: 09 de 10 de 2012.] http://mmlabold.ceid.upatras.gr/courses/AIS_SITE/files/3%5CA%20Survey%20of%20Web%20Services%20Security.pdf. 978-3-540-22054-1.

12. Lucena López, Manuel J. *UNED*. [En línea] Mayo de 2003. [Citado el: 1 de 11 de 2012.] http://www.uned.es/413042/material/Criptografia.pdf.

13. Menezes, Alfred J. *Handbook of applied cryptography*. [En línea] Octubre de 1996. [Citado el: 05 de 11 de 2012.] http://cacr.uwaterloo.ca/hac/.

14. Montenegro, L. *Seguridad de la Información: Más que una actitud, un estilo de vida.* [En línea] MVP. [Citado el: 12 de 10 de 2012.] http://www.microsoft.com/conosur/technet/articulos/seguridadinfo/.

15. Kimaldi. *Controlador de Acceso*. [En línea] [Citado el: 15 de noviembre de 2012.] http://www.kimaldi.com/empresa/quienes_somos.

16. CNX ANISTER . *Distribuidor global de soluciones de seguridad.* [En línea] [Citado el: 21 de noviembre de 2012.] http://www.anixtersoluciones.com/latam/cl/seguridad/18043/migre_hacia_la_mayor_tecnologia_en_control_de_accesos_es.htm.

17. AMAG TECHNOLOGY. *Javelin form AMAG technology*. [En línea] [Citado el: 20 de noviembre de 2012.] http://www.securityinfowatch.com/product/10327856/amag-technology-javelin.

18. SmartCard Systems S.A. *Sistemas de control de acceso y software de control de acceso, HID.* [En línea] 2 de Enero de 2013. [Citado el: 15/12/ 2012.] http://www.scssa.com.ar/control-de-acceso.htm.

19. Arquero Sistema corporativo. *Software de control y seguridad. Control de acceso y control de horario.* [En línea] 15 de Enero de 2013. [Citado el: 16/ 12/2012.] http://www.sci-spain.com/root.php?modulo=controlAcceso.

20. Binary Ingeniería y Software S.L. *Binary Ingeniería y Software Sistemas de control de acceso.* [En línea] 14 de Febrero de 2013. [Citado el: 18 de Diciembre de 2012.] http://www.binaryis.com/ControlAccesos.aspx.

21. GSInnova. *Rational Rose Enterprise*. [En línea] [Citado el: 1 de diciembre de 2012.] http://www.rational.com.ar/herramientas/roseenterprise.html.

22. Visual Paradigm. *UML, BPMN y Database Tools for Software Development.* [En línea] [Citado el: 2 de diciembre de 2012.] http://www.visual-paradigm.com/.