Módulo para el control de acceso físico para la plataforma OpenERP.

Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas.

Autor:
Adysmarys Vergara León

Tutores:
Ing. Jenny Crespo Cabezas
MSc. Alexander Hernández Chapman

"La Habana, junio del 2014"
"Año 56 de la Revolución"

Declaración de Autoría.

Declaro que soy la única autora del presente trabajo que lleva como título: Módulo para el control de acceso físico para la plataforma OpenERP y autorizo al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio

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

Adysmarys Vergara León _____

Ing. Jenny Crespo Cabezas _____

MSc. Alexander Hernández Chapman _____

Datos de Contacto.

Síntesis de los tutores:

- Jenny Crespo Cabezas: jcrespo@uci.cu
- Alexander Hernández Chapman: alx@uci.cu

Agradecimientos.

A mi Marichú, la estrella que siempre ha estado conmigo.

Le agradezco especialmente a mi mamá, que ha sido más que mi amiga, es todo lo hermoso y extraordinario que tengo en mi vida. Por darme todo el apoyo y el amor que siempre he necesitado.

A mi pollito y a mi segundo papá Gerardo por aguantar mis malcriadeces.

A mis tías Maritza, Leonor, a mis primos Mcdiel y Marlon, los más lindos del mundo, los adoro. A mi hermana Karelia y a mis hermanos postizos Yubiel, Yoan e Ignacio por brindarme su cariño.

Les agradezco a mis antiguas compañeras, Aylen, Anays, Adais, Yaricel, Sealis, Dayana, Amelia, Mabel, Lieny, Sairenis, y a las nuevas Laura, Gloria, Sucel, Greisy por brindarme su hermosa amistad. En especial a los hermanitos que tengo, Tudis, Yosel Lázaro, Cosita Aynel y Peluche.

A profesoras y profesores que me han enseñado mucho y se han convertido para mí en apoyos incondicionales, Yaima, Yisel, Emilio, Alexander a última hora pero llego. Otros profesores que aunque no estén presente se quedaron en mi corazón por el modo de impartirme las clases, María Cristina, Briseis, Zumeta.

A todos los que han compartido conmigo momentos inolvidables y los no tan inolvidables también.

Dedicatoria.

Dedico este trabajo a mi abuela, mi inolvidable Marichú, que aunque no esté presente es muy importante en mi vida, siempre estará en mi corazón. Te amo mimi.

Resumen

La plataforma OpenERP, es un sistema que gestiona todas las áreas de una empresa, ejemplo: Logística, Distribución, Marketing, Ventas, Finanzas, Recursos Humanos, Contabilidad, entre otras. Además con la parametrización adecuada y de manera estándar permite personalizar todos los flujos de trabajo que conforman los procesos empresariales.

Actualmente la plataforma no cuenta con funcionalidades que se encarguen de realizar el control de acceso físico a las personas que acceden o trabajan en locales u oficinas de una empresa. La ausencia de procesos que garanticen dicha seguridad provoca la ineficiente protección de todos los medios que resguardan las entidades.

A partir del análisis de los sistemas que realizan el control de acceso físico, a nivel nacional e internacional, se determina que ninguno puede integrarse a la plataforma por las diferentes características que presentan y no cumplir con los requisitos necesarios que requiere la solución OpenERP.

En el presente trabajo, que tributa al Centro de Identificación y Seguridad Digital, propone un módulo que permite realizar el control de acceso físico desde la plataforma OpenERP, el cual fue guiado por un enfoque ágil usando la metodología Programación Extrema (por sus siglas en inglés, XP) y utilizando el lenguaje de programación python sobre el entorno integrado de desarrollo OpenObject.

Palabras clave: control de acceso físico, módulo, OpenERP, OpenObject, plataforma, python.

Índice de Contenido.

Índice de Figuras.

Índice de Tablas.

Introducción

El progreso de la humanidad se sustenta en un desarrollo tecnológico sin precedentes. Paralelo a ello, la evolución de las tecnologías en Cuba se hace cada vez más beneficiosa para el avance de nuestra sociedad. El perfeccionamiento y adelanto de dichas tecnologías proporciona una estructura más organizada en aras de lograr una mejor eficiencia en el funcionamiento de las empresas, centros e instituciones con que cuenta la revolución.

El empleo de las Tecnologías de la Informática y las Comunicaciones (TIC) para la gestión de procesos empresariales es elevado. En este contexto se destaca el *Enterprise Resource Planning* (por sus siglas en inglés, ERP) utilizado en la planificación de recursos en las empresas. El ERP se introduce en una enorme plataforma que cubre todas las secciones de una empresa facilitando el trabajo con los flujos de actividades que en ellas existan.

La plataforma OpenERP (López, 2010) donde se asocia el término *Open* con el de software open source[1] o código abierto; es un completo sistema de gestión empresarial que se puede utilizar en la mayoría de las áreas en una empresa. Ejemplo de las áreas son: Logística, Distribución, Marketing, Ventas, Finanzas, Recursos Humanos (RRHH), Contabilidad, entre otras; demostrando así que puede gestionar una empresa de manera estándar en todos sus departamentos. También, con la parametrización adecuada, puede llegar a personalizar todos los flujos de trabajo de una institución.

La utilización de la plataforma OpenERP en las instituciones cubanas traería consigo las ventajas antes mencionadas, lo cual permitiría facilitar el trabajo del personal y gestionar todo el proceso empresarial desde un único sistema. Cumple con la política propuesta por el país de utilizar herramientas o sistemas que sean totalmente de código libre (Ibarra, 2012).

OpenERP no cuenta con funcionalidades que permitan llevar el control de acceso físico de las personas que acceden a las áreas, locales u oficinas de una empresa. La ausencia de un módulo capaz de realizar dicho control provoca las siguientes deficiencias en las entidades donde se encuentre desplegada dicha plataforma:

- No se realiza el control automatizado de las personas que acceden a áreas o locales, lo cual, de llevarse manualmente, puede implicar violaciones en el acceso: Al no llevarse el

[1] El significado de dicho término hace referencia al hecho de software *OpenSource* (OSS), programas de código abierto, es decir, su código fuente está disponible públicamente.

control automatizado del personal, puede que la persona encargada de restringir el acceso permita que alguien sin autorización acceda al local sin registrarse.

- No se restringe el acceso a los medios (computadores, documentos, información, entre otros) existentes en un local o área determinada: Como no se controla el acceso a los locales, puede que cualquier persona sin la autorización requerida acceda a los medios mencionados o los sustraiga del local, sin que quede constancia del hecho.
- Resulta muy difícil consultar, manualmente, la información para determinar quiénes y cuándo accedieron a un local: Si los registros de acceso a las áreas de la empresa están anotados en papel es engorroso buscar la información referente al acceso de un usuario a un local con fecha y hora determinada.
- No se lleva el control de la información real necesaria de los trabajadores que acceden a un local determinado: Cuando un trabajador accede a un local, al identificarse, puede que los datos personales, proporcionados por él, no sean auténticos.

Lo anterior implica la necesidad de utilizar terceras herramientas que realicen el control de acceso a personas, limitando así las potencialidades de uso de la OpenERP como solución integral.

Teniendo en cuenta la problemática expuesta anteriormente, se identifica como **problema de la investigación**: ¿Cómo automatizar el control de acceso físico de personas a locales en las instituciones donde se encuentra desplegada la plataforma OpenERP?

Se define como **objeto de estudio** el proceso de control de acceso físico de personas en las instituciones. Definiéndose como **campo de acción** los sistemas o herramientas informáticas que realizan el control de acceso físico a personas.

Para dar respuesta al problema planteado se traza el siguiente **objetivo general:** implementar un módulo, para la plataforma OpenERP, que permita realizar el control del acceso físico de personas a locales de una institución.

Teniendo en cuenta el objetivo general identificado anteriormente se desglosan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación caracterizando los sistemas de control de acceso físico existentes a nivel nacional e internacional, identificando además las tecnologías, metodologías y herramientas a utilizar para el desarrollo de la investigación.

- Realizar la documentación acorde a la metodología seleccionada en la investigación para el diseño del módulo de control de acceso físico de personas.
- Implementar el módulo de control de acceso físico de personas e integrarlo a la plataforma OpenERP.
- Realizar pruebas de software para validar el correcto funcionamiento del módulo de control de acceso físico.

Las **tareas** a desarrollar para cumplir el objetivo general de la investigación son las siguientes:

- Definición de los aspectos teóricos referentes a los sistemas de control de acceso físico.
- Análisis del estado del arte de los sistemas de control de acceso físico.
- Identificación de las tecnologías, metodología y herramientas a utilizar para el desarrollo del módulo.
- Especificación de los requisitos del software.
- Realización de las historias de usuarios.
- Implementación de las funcionalidades del módulo de control de acceso físico.
- Realización de pruebas funcionales continúas.
- Realización de pruebas de aceptación y pruebas de unidad al módulo propuesto.

Para ampliar estas tareas de investigación se utilizaron los siguientes **métodos científicos**:

Métodos teóricos:

- **Histórico-lógico**

Se utilizó en el estudio de los antecedentes de la plataforma OpenERP, la evolución y el desarrollo que han tenido los sistemas de control de acceso físico, así como también valorar las tendencias actuales de la plataforma, por ejemplo, los procesos de control de personas a locales de una institución, funcionalidad que no presenta.

- **Analítico-sintético**

Análisis de las bibliografías que permitieron la extracción de las principales características de la plataforma OpenERP y la investigación de los elementos fundamentales que presentan los procesos de control de acceso físico. Posteriormente se resume e incorporan las características obtenidas que proporcionen la integración de los mismos a la plataforma.

- **Modelación**

Permite crear los diagramas correspondientes al diseño del software, en correspondencia con la metodología identificada, de manera que contribuyan a comprender el funcionamiento de la solución propuesta.

Métodos empíricos:

- **Entrevista**

Permitió obtener la información necesaria relacionada con los problemas existentes en el control de acceso físico en una entidad y obtener documentación de las dificultades que pueden presentar si no se lleva un monitoreo de los flujos de actividades en una empresa; en este caso la entrevista se realizó en la Universidad de las Ciencias Informáticas (UCI) específicamente en Centro de Identificación y Seguridad Digital (CISED) donde se desarrolló un sistema de control de acceso físico a los laboratorios de producción que tiene el centro.

- **Observación**

Observación del objetivo fundamental, el monitoreo del acceso físico en una institución, o sea un control físico de las personas que acceden o trabajan en las áreas, locales u oficinas de una empresa. Obtener información real sobre la ejecución de los procesos del acceso físico que estén presentes en el negocio para la formulación del problema a resolver.

Justificación de la investigación:

La relevancia de la investigación se concentra en el valor agregado que le aporta a la plataforma OpenERP un módulo que permita el control de acceso físico de personas a locales de una institución. El módulo propuesto da la posibilidad, desde la plataforma OpenERP, de llevar un informe de cada actividad de entrada o salida comprobando el acceso a las instalaciones por el personal autorizado, así como velar por la protección de los activos y medios que resguardan estas empresas.

El presente trabajo de diploma se divide en tres capítulos estructurados de la siguiente forma:

Capítulo I: Fundamentación teórica. En este capítulo se realiza un estudio del estado del arte a nivel nacional e internacional de los sistemas de control de acceso físico, así como se puntualizan

los principales conceptos relacionados con la seguridad y los propios sistemas. Se identifican características de la plataforma OpenERP, controles de seguridad, sistemas de control de acceso y el modelo de control de acceso, además se definen e identifican las tecnologías, metodología y herramientas a utilizar durante el desarrollo del módulo propuesto.

Capítulo II: Análisis y diseño de la solución propuesta. En este capítulo además de realizarse una descripción general de la solución propuesta y su funcionamiento, define los principales aspectos relacionados con su diseño. Se realiza la captura de los requisitos funcionales y no funcionales, identificando igualmente la arquitectura que organice la lógica del módulo. También, se especifican los patrones de diseño a utilizar y los artefactos derivados de la metodología de desarrollo de software seleccionada.

Capítulo III: Implementación y Pruebas. Este capítulo aborda aspectos relacionados con la implementación del componente. Además de realizar las pruebas unitarias y de integración al módulo, dirigidas al código, para verificar que responda a un correcto funcionamiento, también se construyen los casos de prueba de aceptación en base a los requisitos definidos inicialmente para el desarrollo de la aplicación.

Capítulo 1: Fundamentación Teórica.

Introducción

En el presente capítulo se elabora el marco teórico de la investigación. Para ello se realiza un estudio de las tendencias actuales en los sistemas de control de acceso físico y se exponen los conceptos fundamentales que aportan información relevante a la base del progreso de la solución propuesta. Además se justifica la selección de las herramientas, metodologías y tecnologías a utilizar para el desarrollo de la misma.

1.1 Principales conceptos.

A continuación se presentan definiciones y conceptos que ayudarán a la comprensión de los temas a desarrollar en la investigación.

1.1.1 Sistema de gestión ERP

Un ERP, también conocido como sistemas integrales de empresas o sistemas integrados de gestión, es un software de gestión empresarial capaz de modelar y automatizar la mayoría de procesos de una empresa. Es utilizado por todos los departamentos de la organización resultando una herramienta eficaz en la toma de decisiones y planificación de los recursos de una empresa.

ERP principalmente, satisface las necesidades de información de una empresa y es un soporte en la dirección de un negocio y en el control del cumplimiento de sus objetivos. O sea, se trata de la integración de una base de datos, una aplicación y una interfaz de usuario en el que se recopila todo el volumen de información que se manipula en una institución. Ayuda a trabajar de forma eficaz y rápida debido a que la información se encuentra disponible y actualizada al instante. Además, evita la duplicidad de datos al tener la información unificada y ordenada en un solo lugar, en una base de datos compartida.

Como consecuencia de esta integración, los ciclos productivos se ven reducidos, agilizando los diferentes tipos de trabajo de cada usuario del sistema con la disminución de tiempo en tareas repetitivas y el aumento de la comunicación entre todos los departamentos de la empresa (López, 2010).

1.1.2 Plataforma OpenERP v7.0

La plataforma OpenERP es un completo sistema de gestión empresarial, cubre las necesidades de muchas áreas; como por ejemplo: Contabilidad, Finanzas, Ventas, Recursos Humanos (RRHH), Compras, Proyectos y Almacén.

OpenERP permite la gestión dinámica de los distintos procesos de negocio de manera gráfica e intuitiva, gracias a su potente sistema de generación de flujos de trabajo, el sistema brinda la posibilidad de editar y modificar flujos de trabajo directamente desde la pantalla. La plataforma es

un software de ERP bajo la licencia GPL[2] inicialmente desarrollado en Bélgica. OpenERP está principalmente enfocado a gestionar una empresa de manera estándar en todos sus departamentos. Puede llegar a personalizar todos los flujos de trabajo, aunque dispone de módulos como gestión de proyectos o estadísticas, más habituales en empresas de mayor envergadura, implementa funcionalidades como Ventas, Compras, Contabilidad y Tesorería que también son módulos importantes (Colombia Python, 2014).

Una de las características técnicas más relevantes de OpenERP es que las bases de datos de información están gestionadas mediante PostgreSQL (Martinez, 2014), un sistema de gestión open source o código abierto orientado a objetos, y que ha sido programado con el lenguaje de programación llamado python (Colombia Python, 2014).

OpenERP se encuentra en un estado funcional sobre Linux[3] y Windows[4], internamente usa un modelo de Flujos de Trabajo (en inglés, *WorkFlow*), con arquitectura en tres capas (interfaz, lógica de negocio y persistencia). Su interfaz ofrece la posibilidad de hacer modificaciones y adaptaciones en cualquier pantalla; añadir y modificar campos, formularios o informes y todo ello, sin necesidad de programar (López, 2010).

1.1.3 Controles de seguridad

El control de acceso es una de las actividades más importantes de la arquitectura de seguridad de un sistema. Un control es lo que permite garantizar que cada aspecto, que se valoró con un cierto riesgo, quede cubierto y auditable, pero no se debe confundir la actividad de control de accesos con autenticación, esta última tiene por misión identificar que verdaderamente “sea, quien dice ser”. El control de acceso es posterior a la autenticación y debe regular que el usuario autenticado, acceda únicamente a los recursos sobre los cuales tenga derecho.

---

[2] GPL (General Public License): Licencia que permite instalar y utilizar un programa GPL sin limitación en número de ordenadores. Además, permite distribuir y modificar el programa para adaptarlo a las necesidades del usuario sin ningún tipo de restricción. La única obligación que conlleva esta licencia es la de facilitar el programa binario y el código fuente. Una licencia GPL no conlleva costes de adquisición, la documentación del programa es abundante, la implementación es flexible y la aplicación permite la perfecta personalización en función del uso que se le quiera dar.

[3] Linux: Sistema operativo de software libre. No es necesario comprar una licencia para instalarlo y utilizarlo en un equipo informático. Es un sistema multitarea, multiusuario, compatible con UNIX, y proporciona una interfaz de comandos y una interfaz gráfica, que lo convierte en un sistema muy atractivo y con estupendas perspectivas de futuro.

[4] Windows: Sistema operativo basado en ventanas, proviene de nombre de una familia de sistemas operativos desarrollados y vendidos por Microsoft.

La seguridad computacional a menudo se divide en tres categorías distintas, control físico, control técnico y control administrativo, estas definen los objetivos principales de una implementación para una seguridad apropiada (Inc. Red Hat, 2005).

Control físico: El control físico es la implementación de medidas de seguridad en una estructura definida usada para prevenir o detener el acceso no autorizado a material confidencial. Algunos ejemplos son las cámaras de circuito cerrado, los sistemas de alarmas técnicos o de movimiento, y la biométrica que incluye huellas digitales, voz, rostro, iris, escritura a mano y otros métodos para reconocer a los individuos que tienen acceso a una institución o centro determinado (Inc. Red Hat, 2005).

Control técnico: Los controles técnicos utilizan la tecnología como una base para controlar el acceso y uso de datos confidenciales a través de una estructura física y sobre la red. Los controles técnicos son mucho más extensos en su ámbito e incluyen tecnologías tales como la encriptación, tarjetas inteligentes, autenticación a nivel de la red, listas de control de acceso (por sus siglas en inglés, ACL) y el software de auditoría de integridad de archivos (Inc. Red Hat, 2005).

Control administrativo: Los controles administrativos definen los factores humanos de la seguridad. Incluye todos los niveles del personal dentro de la organización y determina cuáles usuarios tienen acceso a qué recursos e información (Inc. Red Hat, 2005).

Los Sistemas de Control de Acceso conceden permisos a usuarios o grupos de acceder a objetos, tales como ficheros o impresoras en la red. El control de acceso está basado en tres conceptos fundamentales: identificación, autenticación y autorización. Incluye autenticar la identidad de los usuarios o grupos y autorizar el acceso a datos o recursos. Son esenciales para proteger la confidencialidad, integridad y disponibilidad de la información, el activo más importante de una organización, pues impide que los usuarios no autorizados accedan a los recursos o datos que existen (Inc. Red Hat, 2005).

La propuesta de solución para la posterior integración con la plataforma OpenERP, debe utilizar los elementos de la categoría del control físico, dentro de las tres que se definen. Además se identifica con algunos aspectos del control técnico y administrativo.

1.1.3.1 Control de acceso físico

Esta solución permite el control de los puntos estratégicos de una compañía mediante equipos que verifican la identidad de las personas en el momento de ingresar a las instalaciones. Este tipo de control maneja políticas totales o parciales de acceso, mantiene control de tiempo de las personas que realizan transacciones mediante un manejo avanzado de credenciales que permite

controlar, limitar, monitorear y auditar el acceso físico. Este tipo de sistema es ideal para organizaciones que desean controlar una única área restringida o múltiples puertas de acceso (Technology Integrator, 2014).

1.1.3.2 Modelo de control de acceso basado en roles (por sus siglas en inglés, RBAC)

El control de acceso es un aspecto fundamentan de la seguridad. Permite fundamentalmente preservar la confidencialidad, disponibilidad e integridad de la información. Estas categorías se describen a continuación:

- La confidencialidad se refiere a la necesidad de mantener la información segura y privada. Esta categoría puede incluir desde Secretos de Estado a Confidencial Memorandos, Información Financiera, y la Información de Seguridad como contraseñas.
- La integridad se refiere al concepto de protección de la información, de ser indebidamente alterada o modificada por usuarios no autorizados.
- La disponibilidad se refiere a la noción de que la información está disponible para utilizarse cuando sea necesario (Ferraiolo, 2007).

Las motivaciones principales detrás de RBAC son la capacidad de detallar y hacer cumplir las políticas de control de acceso específicas de la empresa, o sea, hacer más eficiente el proceso engorroso de la gestión de autorización. RBAC representa un importante avance en la flexibilidad y el detalle del control de los existentes estándares de DAC[5] y MAC[6]. Según se define en el TCSEC[7] y comúnmente implementado, DAC es un acceso político de control y el mecanismo que permite a los usuarios del sistema permitir o no permitir que otros usuarios tengan acceso a los objetos bajo su control. La política TCSEC DAC define: Un medio para restringir el acceso a objetos basado en la identidad de los sujetos o grupos, o ambos, a la que pertenecen. Los controles son discrecionales en el sentido de que un sujeto con un permiso de acceso seguro es capaz de pasar de un permiso (quizás indirectamente) a cualquier otro tema, a menos que sea restringido por MAC (Ferraiolo, 2007).

---

[5] Control de acceso discrecional (por sus siglas en inglés, DAC): Es una especie de control de acceso definido por los Criterios de Trusted Computer System Evaluation (TCSEC): "como una forma de restringir el acceso a objetos basados en la identidad de los sujetos o grupos a los que pertenecen.

[6] Control de acceso al medio (por sus siglas en inglés, MAC): Es el conjunto de mecanismos y protocolos a través de los cuales varios "interlocutores" (dispositivos en una red, como ordenadores, teléfonos móviles, etc.) se ponen de acuerdo para compartir un medio de transmisión común (por lo general, un cable eléctrico u óptico o, en comunicaciones inalámbricas, el rango de frecuencias asignado a su sistema).

[7] *Trusted Computer System Evaluation Criteria* (TCSEC): Es un Departamento del Gobierno de los Estados Unidos de Defensa (DoD) y su norma establece algunos requisitos básicos para la evaluación de la eficacia en los controles de seguridad informática integrados a un sistema de cómputo. El TCSEC se utilizó para evaluar, clasificar y seleccionar los sistemas informáticos que se están considerando para el procesamiento, almacenamiento y recuperación de información sensible o clasificada.

DAC, como su nombre lo indica, permite la aprobación y revocación de accesos, o sea permisos que se dejan a discreción, de los usuarios individuales. El mecanismo DAC permite a los usuarios conceder o revocar el acceso a cualquiera de los objetos bajo su control sin la intervención de un administrador del sistema.

Con RBAC, las decisiones de acceso se basan en los roles de usuarios individuales que tiene una organización. Esto incluye la especificación de las funciones, responsabilidades y calificaciones (Ferraiolo, 2007). Por ejemplo, las funciones de una persona asociada con la plataforma se basan en el rol que asume, en este caso el de administrador.

Los usuarios normalmente no pueden pasar sus permisos a otros usuarios. Por ejemplo, el técnico encargado de controlar el acceso físico de las personas tiene el permiso, otorgado por el administrador, de monitorear todo el flujo del personal que tiene la autorización de acceder a los locales u áreas de una empresa.

Desde una perspectiva empresarial RBAC tiene el potencial de ofrecer varios beneficios. Esto incluye una mayor productividad administrativa en la realización común de funciones para la gestión de autorización. Estas funciones administrativas se refieren a la asignación de permisos para que un nuevo usuario acceda a los recursos (tanto los nuevos usuarios como los nuevos recursos), la revisión o eliminación selectiva de accesos que ya no son necesarios (y potencialmente peligrosa) con respecto a un usuario que cambió de asignación o de trabajo, y la integridad y la inmediatez de la eliminación de permisos en el caso de separación de un usuario de la empresa. Estas mismas características han demostrado su capacidad para aumentar la productividad al reducir el tiempo de inactividad entre los eventos administrativos, cuando la empresa se vería privada de la productividad durante el período que el usuario es incapaz de acceder a los recursos del sistema (Ferraiolo, 2007).

Para la propuesta de solución se identifica este modelo de control de acceso como el indicado para permitir mayor productividad administrativa, o sea asignación de permisos para que los usuarios, previamente autorizados, accedan a los recursos de la empresa.

1.1.4 Estado del arte de los sistemas de control de acceso físico en la actualidad

A continuación se expone una breve descripción de los sistemas de control de acceso físico a nivel nacional e internacional encontrados en la bibliografía. Luego se realizará un análisis crítico de cada uno en base a si pueden o no integrarse en la plataforma OpenERP.

1.1.4.2 Sistemas de control de acceso físico a nivel internacional.

CS-Access

CS-Access es un software de control de acceso de personal especialmente diseñado para controlar los accesos y garantizar la seguridad de las instalaciones de las pequeñas y medianas empresas de cualquier sector de actividad. *CS-Access* es un completo sistema de seguridad que permite la apertura de puertas e impedimentos de paso mediante tecnología de huella digital, mano, tarjeta de proximidad, tarjeta magnética o PIN, o una combinación de ellas (Sistemas Terminales y Software, 2010).

Software de Control de Acceso Suprema *BioStar*

BioStar es un software de administración que funciona sobre la plataforma Microsoft Windows. El sistema de control de acceso *BioStar* se basa en conectividad de IP[8] y seguridad biométrica. El software habilita los dispositivos de Control de Acceso Suprema para realizar funciones de control de acceso exhaustivo, incluyendo administración de usuarios, administración de dispositivos, control de puertas, zonas, monitoreo en tiempo real, entre otras (Software de Control de Acceso *Biostar*, 2014).

Software de Control de Acceso *AMADEUS 5*

Amadeus5 es un sofisticado software de control de acceso y administración de alarmas que permite, de forma centralizada y en tiempo real, controlar el flujo de empleados y visitantes en las dependencias de una empresa. Dicho sistema transforma las instalaciones en un edificio inteligente. Por ejemplo, el pase de una tarjeta en un lector apagará automáticamente las luces y la calefacción en cualquier área designada, permitiendo así el ahorro de energía (Software de Control de Acceso *AMADEUS 5*, 2010).

Software biométrico EP 300 de la empresa *INBIOSYS*

El sistema EP 300 creado para el control de la entrada y salida de los trabajadores de una empresa. *INBIOSYS* utiliza tecnología biométrica para la autenticación del personal correspondiente a la misma. Al mismo tiempo es un equipo que tiene capacidad de 2000 huellas digitales, almacenando hasta 50000 registros, además posee un control de tiempos y asistencia que se puede instalar en las oficinas para un mejor control de los empleados, conjuntamente tiene un completo programa en español que le permite configurar los horarios de los trabajadores, emitir reportes y exportar la información en archivos planos a otros programas de nómina (INBIOSYS Biometria, 2014).

---

[8] Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP, que corresponde al nivel de red del Modelo OSI. Permite realizar una comunicación utilizando una red IP, ya sea mediante red de área local o a través de *Internet.*

Software biométrico VF 30 de la empresa *INBIOSYS*

El sistema VF 30 controla el acceso de personas, permitiendo la apertura de puertas y el control de tiempos-asistencias con la huella digital. Fue desarrollado para la seguridad de pequeñas y medianas empresas. Al mismo tiempo realiza la integración de la identificación dactilar, de los sistemas RFID[9], las alarmas, los tiempos de atención y las funciones de control de acceso con una elegante apariencia y calidad confiable. Posee una alarma musical de alta calidad, pantalla en varios idiomas, interfaz de usuario amigable y comunicación adecuada para la gestión de datos en diferentes entornos. Incluye un software funcional de gestión, compatible con varios tipos de base de datos y con la zona horaria (*INBIOSYS* Biometria, 2014).

Software de Control Personal *Wapa*

El software *Wapa* fue creado por la empresa *DokkoGrou*[10]*.* Es un software rápido, práctico y completo para obtener toda la información que se necesita sobre cualquier empleado. El mismo facilita el monitoreo de las entradas y salidas del personal de la empresa de forma sencilla y segura mediante la utilización de tarjetas o llaveros con códigos únicos e irrepetibles. Además utiliza unidades lectoras por radio-frecuencia que permiten que el personal quede registrado con sólo aproximar la tarjeta a la unidad central, evitando de esta manera errores de lectura causados por desgastes o malos usos. Conjuntamente tiene como beneficio el acceso al sistema mediante una computadora con conexión a *Internet*[11] desde cualquier parte del mundo, permitiendo así que se pueda monitorear y administrar el personal de la empresa estando fuera de la misma (DokkoGroup, 2014).

1.1.4.3 Sistemas de control de acceso físico a nivel nacional

Biomesys Control de Asistencia

Sistema que aprovecha las bondades de las tecnologías biométricas para registrar los eventos de asistencia en una organización por medio de la identificación de los empleados y de la autenticación de la identidad mediante un sensor biométrico de huellas dactilares. Asimismo, con la captura de identificaciones biométricas únicas, se convierte en un generador de datos altamente confiable por su bajo o casi nulo nivel de vulnerabilidad por la suplantación de identidad. *Biomesys* tiene como características distintivas no establecer limitaciones de implementación asociadas al tipo de organización; integrar de forma rápida con diferentes medios de autenticación como

[9] *Radio Frequency Identification* (por sus siglas en inglés, RFID): Es un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID.

[10] *DokkoGrou*: Empresa argentina que ofrece diversidades de productos tecnológicos.

[11] *Internet*: Conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.

escáneres biométricos, credenciales de bandas magnéticas, tarjetas de códigos de barras y proximidad; posee un módulo de captura biométrica para el control de las entradas y las salidas del personal (DATYS Tecnología y Sistemas, 2011).

Frontpas

Es una solución integral para la gestión y control de la frontera. El sistema integra el registro, control y vigilancia de pasos fronterizos legales trabajando con estándares de seguridad y calidad internacionales. *Frontpas* provee una plataforma tecnológica que facilita el control y gestión del tránsito de personas en las fronteras legales y los flujos en condiciones de estricta seguridad, permitiendo contrarrestar, según el interés nacional, el tráfico ilegal de personas, mercancías y combustibles. Gestiona el proceso de chequeo migratorio en general desde cualquier punto fronterizo; captura y utiliza la información multibiométrica (rostro y huella) para detectar y evitar la suplantación de identidad, además verifica la identidad de la persona contra listas de control (donde se registran aspectos o rasgos de una persona). Permite además el chequeo de listas de pasajeros de vuelos (API[12]) previo a la llegada de los viajeros (DATYS Tecnología y Sistemas, 2011).

XymaSafeVision

Es un sistema de video protección profesional basado en la tecnología IP. El mismo se integra a los diversos equipamientos de cómputo, videocámaras y servidores para cámaras analógicas. También tiene incorporado algoritmos de reconocimiento de patrones que permiten identificar y verificar las matrículas de los autos, definir perímetros para la detección de movimiento o cantidad de objetos, entre otras. Contiene una gran capacidad de manejo de usuarios y cámaras, puede realizar grabaciones de forma manual, programada, continua, por detección de movimiento o por eventos de alarmas predefinidos. *XymaSafeVision* admite interactuar con los servicios de mapas y planos de las plantas de las instalaciones en apoyo a la administración y utilización del sistema. Del mismo modo posee herramientas para la gestión del ancho de banda de la red y la calidad de las imágenes a través de la personalización de parámetros de video de cada canal y para cada función (DATYS Tecnología y Sistemas, 2011).

Sistema de control de acceso en el Centro de Identificación y Seguridad Digital (CISED)

Este sistema controla el acceso de las personas a los laboratorios destinados a los proyectos productivos, donde se comprueba mediante el documento de identificación (solapín) que los antecedentes del personal estén en la base de datos. Estas identificaciones son verificadas mediante un lector de código de barras, el cual es usado en la universidad. Una de las características más importantes de este sistema es que cada usuario tiene asignado a una

[12] *Advanced Passenger Information System* (por sus siglas en inglés, API): Intercambio electrónico de datos.

dirección IP dentro de los laboratorios productivos, al autenticarse, el sistema reconoce que la persona se ha identificado y hará uso del IP proporcionado; permite además su acceso a la intranet de la universidad e internet.

Sistema de control de acceso a comedores (CONTACC)

Este sistema contiene toda la gestión de control de acceso de los estudiantes, trabajadores y profesores a los comedores de los diferentes complejos alimentarios 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 en cada una de las puertas de los comedores se controla registrando el código de barras, a través de la identificación de cada persona, garantizando que esto tenga lugar una sola vez y por la puerta correspondiente. La gestión de comensales permite a los directivos la asignación de la puerta y el complejo correspondiente a cada persona, brindando reportes sobre los que tuvieron acceso, haciendo seguro y confiable el proceso de identificarlos. Posee una interfaz amigable, se adapta con facilidad a variaciones en la estructura del flujo informativo y permite configurar los tiempos de sincronización de forma manual o automática cada determinado momento.

1.1.5 Análisis de los sistemas de control de acceso físico

Luego del estudio de los sistemas que realizan el control de acceso físico a nivel nacional e internacional queda evidenciado:

Los sistemas *CS-Access, BioStar y Amadeus5* no se pueden integrar a la plataforma debido a que emplean tecnología biométrica que responde a exigencias complejas. Por su parte EP 300, VF 300 *y Wapa*, así como los mencionados anteriormente son sistemas privativos, lo cual va en contra de las política de migración a sofware libre que propone el país.

Los sistemas nacionales no pueden integrarse a la plataforma pues no cumplen con elementos propios que requiere la propuesta de solución para realizar el control de acceso físico. Ejemplo de esto son los sistemas: *Biomesys* Control de Asistencia el que es adaptable a cualquier sistema de lectura y es un generador de datos altamente confiable; Frontpas sólo controla el flujo migratorio o chequeo de frontera en todos los puntos de entrada y salida ya sean terrestres, marítimos o aéreos y *XymaSafeVision* es un sistema de video monitoreo. Ninguna de las funcionalidades brindada por los sistemas anteriores realiza el control y chequeo de la seguridad a partir del código de barras, por lo que no se pueden tener en cuenta para elaborar una propuesta de solución.

Por otro lado el sistema que controla el acceso de personas en CISED necesita autenticarse desde la red. Asimismo, CONTACC brinda reportes haciendo uso seguro y confiable del proceso

de identificar a las personas y configura los tiempos de sincronización de forma manual o automática, pero no puede tenerse en cuenta para elaborar una propuesta de solución debido a que la información que recoge no es a fin con la requerida a la hora de realizar control de acceso que requiere la plataforma.

De lo anterior se resume que ninguno de los sistemas estudiados puede tomarse como propuesta de solución para realizar el control de acceso físico a personas desde la solución OpenERP. Ninguno cumple con los requisitos para la integración de nuevos módulos o funcionalidades a la plataforma, como son lenguaje de programación (python) e IDE de desarrollo *(*OpenObject*)*, por lo que se evidencia la necesidad de implementar un nuevo módulo que sea capaz de realizar el control físico del personal que tiene acceso a una entidad desde la plataforma OpenERP.

Para el desarrollo de la solución propuesta es necesaria la utilización de herramientas y tecnologías, las cuales serán analizadas a continuación.

1.1.6 Propuesta de las herramientas y tecnologías a utilizar

1.1.6.1 Estado del arte de las metodologías de desarrollo

Las metodologías para el desarrollo de un software son marcos de trabajo usados para planificar, controlar y estructurar los procesos de desarrollo en los sistemas de información, por medio de un conjunto de procedimientos, herramientas, técnicas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. A continuación se realiza un análisis de las metodologías ágiles existentes con el propósito de identificar cuál es la indicada para el desarrollo del módulo que controle el acceso físico de personas desde la plataforma OpenERP.

Metodologías ágiles

La filosofía de las metodologías ágiles es dar mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy dinámicos, se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales. Se describe *Extreme Programming* (por sus siglas en inglés, XP) como la metodología ágil más popular en la actualidad, donde es posible la mejora de incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo la aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software (Pressman, 2010).

Scrum

Scrum es una metodología de desarrollo que no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones (Placio, 2006).

Esta metodología aplica de manera regular un conjunto de buenas prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. En *Scrum* se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, la metodología está especialmente indicada para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son dinámicos o poco definidos, la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.
También se utiliza para resolver situaciones en las que no se está entregando al cliente lo que necesita, cuando las entregas se extienden demasiado, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto (Placio, 2006).

Microsoft Solution Framework (por sus siglas en inglés, MSF)

Es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas (Metodologías de Desarrollo de Software, 2004).

MSF tiene las siguientes características:

Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

Escalable: puede organizar equipos tan pequeños entre de 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.

Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.

Tecnología Agnóstica: puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación (Metodologías de Desarrollo de Software, 2004).

XP (*Extreme Programming*)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Además está basada en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy dinámicos, y donde existe un alto riesgo técnico (Douglas, 2012).

El ciclo de vida ideal de XP consiste en seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. Presenta variedad de ventajas como son la programación organizada con menor taza de errores y la satisfacción del programador. Genera requisitos no funcionales e historias de usuarios que describen en detalle cada una de las acciones que se desarrollan. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de *Extreme Programming* sin cubrir los detalles técnicos y de implantación de las prácticas (Douglas, 2012).

A continuación se expone la selección de la metodología de desarrollo de software a utilizar en la implementación del módulo.

Entre las metodologías ágiles anteriormente analizadas se selecciona XP para el desarrollo del módulo de control de acceso físico que integre a la plataforma OpenERP. La metodología identificada además de ser dinámica, flexible y con requerimientos altamente cambiantes, es para un equipo de desarrollo pequeño y garantiza características que la hacen aplicable en ciertos ambientes. Está centrada en potenciar las relaciones interpersonales para el éxito en el desarrollo del software, facilitando un buen clima de trabajo. Tiene como prioridad satisfacer al cliente mediante un desarrollo iterativo e incremental, abierto a los cambios y caracterizado por un código simple, al mismo tiempo que aplica un conjunto de prácticas que harán la entrega del módulo menos complicada y más satisfactoria tanto para los clientes como para el equipo de entrega.

Además responde a una buena documentación para el funcionamiento efectivo y calidad de la arquitectura del software, respecto del modelado y la documentación de la propuesta de solución.

1.1.6.2 Lenguaje de modelado

Al comenzar un software se debe tener en cuenta que el modelado del mismo es de máxima importancia y tiene una notable repercusión en su etapa de diseño e implementación. Esto le proporciona al ingeniero herramientas y prácticas que le permiten "visualizar" el sistema que va a implementar. Para el desarrollo del módulo propuesto se selecciona el siguiente lenguaje de modelado.

1.1.6.3 Lenguaje Unificado de Modelado (UML 2.0)

El lenguaje unificado de modelado (por sus siglas en inglés, UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. Además está controlado por el Grupo de Administración de Objetos (por sus siglas en inglés, OMG) y es el estándar de descripción de esquemas de software. También está diseñado para modelar sistemas orientados a objetos, y tiene utilidad en otro tipo de cuestiones con relación a la programación. Se compone además de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos que genera se utilizan para crear diagramas, que representan alguna parte o punto de vista del sistema (Rumbaugh, 2000).

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar, describe las semánticas esenciales del significado de estos diagramas y símbolos. Además es empleado para modelar distintos tipos de sistemas como son: sistemas de software, sistemas de hardware, y organizaciones del mundo real (Rumbaugh, 2000).

1.1.6.4 Herramienta de modelado Visual Paradigm for UML 8.0

Visual Paradigm es una herramienta UML. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java[13] y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios

[13] Java: Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros (Baeza, 2013).

Visual Paradigm cumple con las políticas actuales de migración a software libre, siendo multiplataforma, de forma tal que facilita la modelación del software independientemente del sistema operativo que se emplee (Baeza, 2013).

Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML asiste con una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite construir todos los tipos de diagramas de clases (Visual Paradigm, 2013).

1.1.6.5 Lenguaje de Programación

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser ejecutados por máquinas computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Wilson, 1993).

A continuación se dará una breve descripción del lenguaje de programacion a utilizar en el desarrollo del módulo propuesto.

1.1.6.6 Python v2.7

Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Es multiparadigma ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma. Además es administrado por la *Python Software Foundation*[14] (por sus siglas en inglés, PSF). Posee una licencia de código abierto, denominada *Python Software Foundation License*, que es compatible con la licencia pública general de GNU[15] a partir de la versión 2.1.1. Permite dividir su programa en módulos reutilizables desde otros programas y viene también con una gran colección de módulos estándar que se pueden utilizar como base de los programas (Python Colombia, 2014).

[14] PSF: Es una organización sin fines de lucro creada el 6 de marzo de 2001 dedicada al lenguaje de programación Python. La misión de la fundación es fomentar el desarrollo de la comunidad Python 2.0. Es responsable de varios procesos dentro de la comunidad, como el desarrollo de Python, la administración de los derechos intelectuales y de obtener fondos.

[15] GNU: Es un sistema operativo Unix-like desarrollado por el Proyecto GNU. Está formado en su totalidad por software libre. Se basa en el núcleo GNU Hurd y tiene como objetivo ser un sistema de software completo compatible con Unix.

OpenERP está escrito con este lenguaje de programación, aunque por motivos de rendimiento el sistema puede generar un código intermedio precompilado (Borrás, 2010).

Se utilizará este lenguaje de programación para la implementación de la propuesta de solución debido a que es el utilizado para desarrollar en la plataforma OpenERP.

1.1.6.7 Entorno Integrado de desarrollo OpenObject

OpenERP está desarrollada sobre la plataforma OpenObject. Es un proceso denominado *Rapid Application Development* (por sus siglas en inglés, RAD) que acelera el ciclo de desarrollo de un software, es de código abierto y programado en python. Además es el framework de desarrollo Modelo-Vista-Controlador (MVC) en el que se basa OpenERP. El MVC es el patrón de arquitectura software que separa la aplicación OpenERP, la interfaz de usuario (GTK *client*) y la lógica de control en tres componentes distintos denominados "Modelo", "Vista" y "Controlador". Por lo tanto, esta plataforma es la que permite a un usuario de OpenERP desarrollar y adaptar su software de forma rápida (López, 2010).

Como características técnicas básicas de OpenObject destacar las siguientes:

- Contiene *Object-Relational Mapping* (por sus siglas en inglés, ORM): se trata de una técnica de programación en la cual adapta los objetos de un lenguaje de programación orientado a objetos y los guarda como si fueran tablas. Dicho de otra manera, tiene como objetivo la conversión del modelo de objetos que maneja una aplicación a base de datos relacional.
- Contiene un motor de *workflow*: se trata de un software encargado de gestionar y ejecutar los procesos de modelado informático. Es capaz de interpretar eventos y actuar sobre ellos de acuerdo con los procesos que hayan sido definidos.
- Contiene un diseñador de informes: capaz de presentar los datos mediante informes.
- Contiene un motor OLAP cube: se trata de una base de datos multidimensional. Recibe este nombre debido a que los datos se almacenan físicamente en un vector multidimensional, lo que nos da a entender que es una ampliación de las dos dimensiones de una hoja de cálculo (López, 2010).

1.1.6.8 Acceso a datos

1.1.6.8.1 Sistema Gestor de Base de Datos PostgreSQL *v*9.1

Para el desarrollo de la solución se trabaja con la versión PostgreSQL 9.1 debido a que es el sistema gestor de base de datos utilizado por la OpenERP versión 7.0. PostgreSQL 9.1 es un

Sistema Gestor de Base de Datos de licencia gratuita. OpenERP soporta tanto bases de datos PostgreSQL como Oracle[16]. Se toma PostgreSQL porque es el componente del ERP que se encarga de guardar la información de forma estructurada y que permite el acceso rápido a los datos.

Por su parte PostgreSQL es un potente sistema gestor de base de datos objeto-relacional de código abierto. Soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Es altamente escalable, tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede administrar. Incluye una biblioteca de funciones estándar con cientos de funciones integradas que van desde las operaciones matemáticas básicas, operaciones con cadena de caracteres para criptografía y compatibilidad con Oracle (microbufer), además PostgreSQL es un sistema distribuido bajo licencia BSD[17] y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otros sistemas gestores de base de datos comerciales (Martinez, 2012).

1.1.6.9 Sistema Operativo Ubuntu

Ubuntu es una distribución Linux basado en Debian GNU/Linux que proporciona un Sistema Operativo (SO) actualizado y estable para el usuario medio, con un fuerte enfoque en la facilidad de uso y de instalación del sistema. Al igual que otras distribuciones se compone de múltiples paquetes de software normalmente distribuidos bajo una licencia libre o de código abierto.

Aplicaciones de Ubuntu: Ubuntu es conocido por su facilidad de uso y las aplicaciones orientadas al usuario final. Las principales aplicaciones que trae Ubuntu son: navegador web Mozilla Firefox, cliente de mensajería instantánea Empathy, cliente de redes sociales Gwibber, cliente para enviar y recibir correo Evolution, cliente y gestor de *BitTorrents Transmission*[18], grabador de discos Brasero, suite ofimática Open Office, y el instalador central para buscar e instalar aplicaciones (Martinez, 2012).

---

[16]Oracle: Es un sistema de gestión de base de datos objeto-relacional, desarrollado por *Oracle Corporation*. Se considera como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

[17]*Berkeley Software Distribution* (por sus siglas en inglés, BSD): Licencia de software otorgada principalmente para los sistemas BSD.

[18] *Transmission*: Es un cliente liviano, gratuito y de código abierto para la red *BitTorrent*. Está disponible bajo la licencia MIT, con algunas partes GPL, y es multiplataforma. Es compatible con los siguientes sistemas operativos: Mac OS X, Linux, NetBSD, FreeBSD y OpenBSD y BeOS. Ahora también está disponible para Windows.

Seguridad y accesibilidad: El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un cortafuego predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema. Para labores o tareas administrativas en la línea de comandos incluye una herramienta llamada sudo (en inglés, SuperUser do), con la que se evita el uso del usuario administrador. Posee accesibilidad e internacionalización, de modo que el sistema esté disponible para tanta gente como sea posible (es.lidocs.org, 2013).

Se propone este Sistema Operativo para el desarrollo de la propuesta de solución debido a que la plataforma OpenERP se instala comúnmente en el mismo, esto se debe a que está incluido en los repositorios Ubuntu y de esta forma se puede descargar e instalar el software en pocos minutos.

1.1.6.10 Tecnologías

Los códigos de barras se han integrado en cada aspecto de nuestras vidas, se localizan en el supermercado, en tiendas departamentales, farmacias, entre otros. Son solo una forma diferente de codificar números y letras usando una combinación de barras y espacios en diferentes medidas.

Código de Barras

Es una disposición en paralelo de barras y espacios que contienen información codificada en las barras y espacios del símbolo. Almacena información, datos que pueden ser reunidos de manera rápida y con una gran precisión. Representan un método simple y fácil para codificación de información de texto que puede ser leída por dispositivos ópticos, los cuales envían dicha información a una computadora como si hubiese sido tecleada (ConbotasSucias, 2012).

Ventajas del Código de Barras:

El código de barras ha sido creado para identificar objetos y facilitar el ingreso de información, eliminando la posibilidad de error en la captura.

- Se imprime a bajos costos.
- Permite porcentajes muy bajos de error.
- Rapidez en la captura de datos.
- Los equipos de lectura e impresión de código de barras son flexibles y fáciles de conectar e instalar.

Beneficios del Código de Barras:

El código de barras es el mejor sistema de colección de datos mediante identificación automática, y presenta muchos beneficios, entre otros:

- Se mejora la exactitud de los datos, hay una mayor precisión de la información.

- Se tienen costos fijos de labor más bajos.
- Se puede tener un mejor control de calidad, mejor servicio al cliente.
- Se mejora la competitividad.
- Se reducen los errores.
- Se capturan los datos rápidamente.
- Se mejora el control de las entradas y salidas.
- El incremento de la velocidad y exactitud en la toma de datos, no lleva a reducir errores, produce un ahorro de tiempo y dinero.

Aplicaciones:

Las aplicaciones del código de barras cubren prácticamente cualquier tipo de actividad humana, tanto en la industria, comercio, instituciones educativas, instituciones médicas, gobierno, entre otras. Por ejemplo:

- Control de material en procesos.
- Control de tiempo y asistencia.
- Control de acceso.
- Control de embarques y recibos.
- Control de documentos y rastreos de los mismos.
- Rastreos preciso en actividades.
- Rastreos precisos de bienes transportados.
- Levantamiento electrónico de pedidos.

Las barras y espacios aparecen impresos en etiquetas de alimentos, paquetes de envío y brazaletes de pacientes. Podría parecer que todas son iguales, pero no es así. Cada tipo de industria tiene una simbología que maneja como su propio estándar (ConbotasSucias, 2012).

A continuación se muestra en las Figuras 1 y 2 las tecnologías utilizadas para la propuesta de solución.

Figura 1. Código de barras 39 (ConbotasSucias, 2012).

Figura 2. Lector de código de barras Voyager (Componentes.com, 2014).

Un lector de códigos de barras o escáner, es un dispositivo que interpreta un código de barras y envía, como si de un teclado se tratara, los números y letras que lo componen.

El dispositivo para la lectura del código de barras antes mencionado es el lector Voyager MS9520. Es un lector rápido, con gran profundidad de campo y velocidad de lectura. Combina las prestaciones del MS 951 y el MS 961 dentro de un mismo escáner para ofrecer mayor flexibilidad al usuario. El Voyager puede operar en el modo "manos libres" cuando se sitúa sobre su soporte. Se necesita solamente la presentación del código, para que el lector realice automáticamente la lectura del documento de identificación. También es programable para lecturas de corto o largo alcance tanto en el modo automático como manual, con lo que incrementa su confiabilidad y productividad (ConbotasSucias, 2012).

1.1.7 Conclusiones Parciales

A partir del estudio de los sistemas de control de acceso físico a nivel nacional e internacional, se concluye que no tienen las características esenciales para integrarse a la plataforma OpenERP, demostrando la necesidad de desarrollar un módulo de control de acceso físico que forme parte de la solución OpenERP. Para el desarrollo de la propuesta de solución se utilizará la metodología XP con Visual Paradigm como herramienta CASE y UML como lenguaje de modelado. Además se empleará python como lenguaje de programación y OpenObject como IDE de desarrollo.

Capítulo 2: Análisis y diseño de la solución propuesta.

Introducción

El presente capítulo contiene las principales características de la solución informática a desarrollar. Se define el modelo de dominio de la aplicación con el propósito de englobar los principales conceptos con los que trabaja el componente. Incluye la captura de los requisitos funcionales y no funcionales, además de conformarse las historias de usuario relacionadas a estos.

En correspondencia con la etapa de planificación del software se puntualizan los planes de entrega, iteraciones y las tareas ingenieriles. También describe la arquitectura candidata del módulo, los patrones de diseño a aplicar y los artefactos más importantes, el diagrama de clases del diseño y las tarjetas CRC correspondientes a cada una de las clases definidas.

2.1 Conceptos fundamentales

A continuación se muestran los conceptos para el análisis y el diseño que permiten un mejor desarrollo de la propuesta de solución.

2.1.1 Descripción del problema.

La plataforma OpenERP está diseñada para cubrir todas los procesos de una empresa, no obstante, carece de funcionalidades para llevar el control de la seguridad en estas instituciones. Para ello se propone el desarrollo del módulo para el control de acceso físico que se integre a OpenERP. El módulo propuesto realizará el control y monitoreo del acceso físico del personal a sus correspondientes locales, verificando la identificación de los mismos; protegiendo así los activos e información que resguarda la entidad.

2.1.2 Propuesta de solución

Modelo de dominio

Algunas veces se observa el modelo conceptual no como un modelo riguroso de información, sino como una herramienta de la comunicación, con el cual se intenta entender los conceptos importantes y sus relaciones. Desde esta perspectiva, eliminar algunas asociaciones que no reclamen estrictamente el criterio de la necesidad de conocer puede originar un modelo inadecuado: no comunica las ideas ni las relaciones más importantes (Larman, 1999). El módulo de control de acceso físico tendrá sus bases en un modelo de dominio, el que permitirá obtener una mejor comprensión del entorno del módulo, mostrando los principales conceptos con los que trabaja, que a su vez describen las clases más importantes dentro del contexto de la aplicación, en la Figura 3 se muestra el modelo de dominio .

Empleado: Persona que tiene la acreditación para el acceso autorizado a las instalaciones de una empresa.

Visitante: Es la persona que va acceder eventualmente a los locales pero no tiene una acreditación autorizada para el acceso.

Registro Manual: Recurso que controla el acceso autorizado a un conjunto de trabajadores o visitantes que necesitan entrar a la empresa.

Empresa: Instalación a la que los empleados y visitantes tienen acceso.

Locales: Locales que tiene la empresa.

Encargado del Registro: Es la persona encargada de registrar la entrada/salida de los empleados que tengan acceso en los locales de la empresa, en el caso de los visitantes emitir un pase de acceso.

Pase de Acceso: Documento que identifica el acceso autorizado a los locales de una empresa de un visitante.

Identificación: Documento que autoriza a un empleado el acceso a los locales de la empresa.

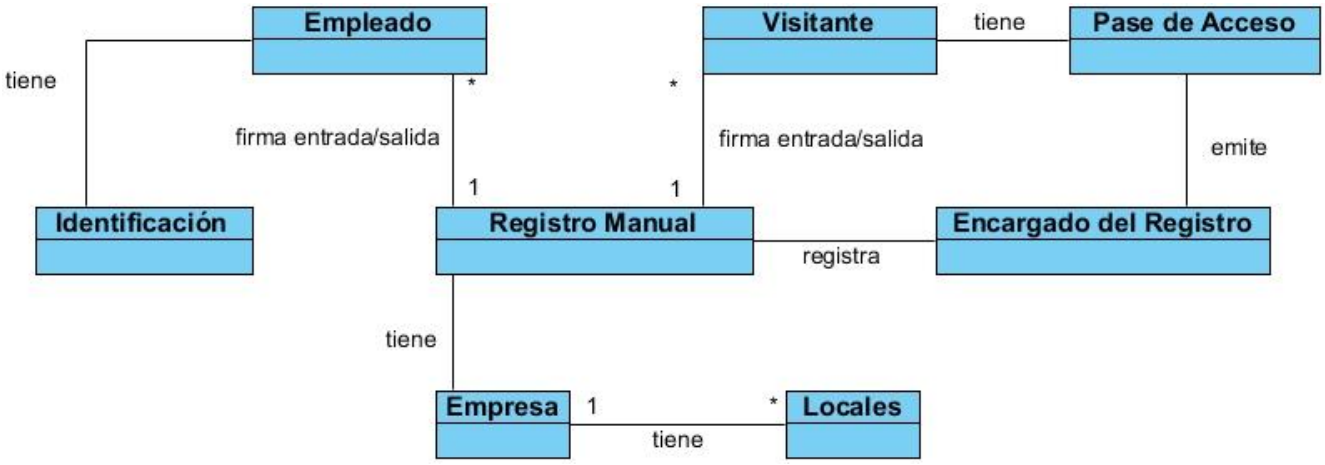


Figura 3. Modelo de dominio.

2.1.3 Especificación de los requisitos del software

2.1.4 Requisitos funcionales

Los requisitos funcionales que describen las funcionalidades que la solución propuesta debe cumplir son:

RF-1 Gestionar locales en el sistema.

a) Crear local.
b) Eliminar local.
c) Modificar local.
d) Mostrar local.
e) Listar locales.

f) Adicionar locales a un empleado.

g) Adicionar a locales los usuarios que pueden registrar el acceso.

RF-2 Asignar usuarios a locales a los que pueden registrar el acceso.

RF-3 Registrar el acceso del personal a locales.

a) Registrar la entrada de empleados a locales.

b) Registrar la salida de empleados a locales.

c) Registrar la entrada de visitantes a locales.

d) Registrar la salida de visitantes a locales.

RF-4 Monitorear el acceso del personal a locales.

a) Listar los empleados que han tenido acceso a un local.

b) Listar los locales a los que un empleado ha tenido acceso.

c) Listar los visitantes que han tenido acceso a un local.

RF-5 Monitorear las infracciones.

a) Mostrar las infracciones en un rango de fecha.

b) Mostrar las infracciones de una persona por un punto de control en un rango de fecha.

2.1.5 Requisitos no funcionales

Son propiedades o cualidades que hacen al producto atractivo, usable, rápido y confiable. Se definieron como requisitos no funcionales:

Requisitos de Software:

RNF-1 El módulo propuesto se ejecutará sobre la Plataforma OpenERP versión 7.0.

Requisitos de Hardware:

RNF-2 Periféricos: Mouse.

RNF-3 Cliente: Lector de código de barras.

RNF-4 Procesador Intel Pentium 4 o superior.

RNF-5 1 GB o más de memoria RAM.

RNF-6 Disco duro de 10 GB o superior.

Restricciones en el diseño y la Implementación:

RNF-7 La implementación del módulo debe realizarse en el lenguaje python v2.7, puesto que la plataforma está implementada en dicho lenguaje.

Requisitos de apariencia o interfaz externa:

RNF-8 La interfaz debe cumplir los requisitos de diseño ya existentes en la plataforma OpenERP, en consecuencia al diseño propuesto por la misma.

Requisitos de seguridad:

RNF-9 Cada usuario, en la aplicación, tendrá acceso a las funcionalidades en función de sus privilegios o niveles de acceso.

2.1.6 Historias de usuarios

Es la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas (Douglas, 2012).

A continuación se definen las historias de usuario que tienen mayor impacto en la solución propuesta, el resto está definido en los anexos.

Tabla 1. HU Asignar a usuarios los locales en los que pueden registrar el acceso.

Historia de Usuario	
Número: HU_4	**Usuario:** Adysmarys Vergara León.
Nombre de Historia de Usuario: Crear Locales en la plataforma.	
Prioridad en negocio: Alta	**Iteración asignada:** 2
Riesgo en desarrollo: Normal	**Puntos estimados:** 5
Programador Responsable: Adysmarys Vergara León.	
Descripción: Cuando el módulo es utilizado por primera vez en una entidad es necesario la creación, en el módulo del sistema, de los locales o áreas con que cuenta la empresa. Además si en la entidad se crea una nueva área es necesario también su adición al sistema. Por cada área que se adicione en el sistema es necesario introducirle un nombre y el listado de empleados que trabajarán en la misma.	
Observaciones:----------	

Tabla 2. HU Asignar usuarios en locales a los que pueden registrar el acceso.

Historia de Usuario	
Número: HU_5	**Usuario:** Adysmarys Vergara León.
Nombre de Historia de Usuario: Asignar usuarios a locales a los que pueden registrar el acceso.	
Prioridad en negocio: Alta	**Iteración asignada:** 2
Riesgo en desarrollo: Normal	**Puntos estimados:** 4
Programador Responsable: Adysmarys Vergara León.	
Descripción: Cuando es necesario que una persona tenga acceso a un local o área se realiza la asignación, desde el módulo, a dichos locales donde tendrá el acceso correspondiente. Por cada nuevo acceso que se registre a un local se guarda, de la persona, el Código de Entrada (solapín). Otra forma de realizar esta funcionalidad es desde el módulo de RRHH, donde en los datos de los trabajadores también se puede asignar uno o varios locales.	
Observaciones:----------	

2.1.7 Planificación

2.1.7.1 Plan de entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente (Letelier, 2010).

El equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de los puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La velocidad del proyecto es utilizada para establecer el número de historias de usuario que se pueden implementar antes de una fecha determinada o el tiempo que tomará implementar un conjunto de historias

Se propone en la Tabla 3 el siguiente plan de entregas para la solución propuesta:

Tabla 3. Plan de entrega.

Entregable	Iteración	Fin de la Iteración
Gestión de empleados y visitantes.	1	Enero del 2014
Utilización de otros módulos de la plataforma.	1	Enero del 2014
Gestión de locales.	2	Febrero del 2014

Asignar usuarios a los locales donde registren el acceso.	2	Marzo del 2014
Registrar entrada de empleados y visitantes.	2	Marzo del 2014
Registrar salidas de empleados.	2	Marzo del 2014
Listar empleados y visitantes que han tenido acceso a un local.	3	Abril del 2014
Listar locales a los que los empleados o visitantes han tenido acceso.	3	Abril del 2014
Mostrar listado de infracciones.	3	Mayo del 2014

2.1.7.2 Plan de iteraciones

El ciclo de desarrollo de software guiado por XP se caracteriza por ser iterativo e incremental, por lo que se realizan varias iteraciones sobre el sistema antes de su fase de producción.

Los elementos que deben tomarse en cuenta durante la elaboración del plan de iteraciones son las historias de usuario no abordadas, la velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero realizadas por parejas de programadores (Letelier, 2010).

Tabla 4. Plan de iteraciones.

Iteración	Historias de Usuarios	Semanas Estimadas
1	Utilización de otros módulos.	4
	Gestión de empleados y visitantes.	
2	Gestión de locales.	4
	Asignar empleados a los locales.	
	Registrar entrada de empleados o visitantes.	
	Registrar salidas de empleados o visitantes.	
3	Listar empleados o visitantes que han tenido acceso a un local.	4
	Listar locales a los que un empleado ha tenido acceso.	
	Mostrar listado de infracciones.	

En la primera iteración se seleccionan algunos de los módulos que tiene implementado la solución OpenERP para hacer uso de los mismos en el control del acceso físico, con esto se gestionan los usuarios a través del módulo de RRHH, que van a interactuar con la entidad, por lo que en la segunda iteración, con la comunicación del módulo establecida, se realiza la gestión de los locales que tiene la entidad y se da comienzo a la asignación de los usuarios. A partir de ahí se hace el registro de entrada y salida de los empleados o visitantes que tengan acceso al centro, mostrando entonces el listado de trabajadores o visitantes que han tenido acceso a los locales y listar los locales a los que un empleado o visitante ha tenido acceso, así como tener control de las incidencias o infracciones cometidas por algún trabajador.

Las tareas de programación o ingenieriles a realizarse en cada una de estas iteraciones se especifican a continuación en la Tabla 5.

Tabla 5. Tareas ingenieriles.

Iteración	Historias de usuarios	Tareas
1	Utilización de otros módulos.	• Selección de módulos que permiten su utilización en el módulo para el control de acceso. • Gestionar los datos correspondientes.
	Gestión de Empleados y Visitantes.	• Definir los empleados que tienen acceso a los locales de la entidad. • Definir los visitantes que tienen acceso a los locales de la entidad. • Definir el responsable del visitante.
2	Gestión de locales	• Crear locales. • Modificar locales. • Eliminar locales. • Mostrar locales.
	Asignar Empleados y Visitantes a los locales	• Designar Empleados a locales gestionados. • Designar Visitantes a locales

		gestionados. • Verificar su acceso en los locales. • Eliminación de la asignación de Empleados a locales donde ya no tengan acceso.
	Registrar entrada de Empleados y Visitantes.	• Verificar que tenga previa identificación para el acceso a la entidad. • Verificar si está en la base de datos de la entidad y tiene acceso al local a donde se dirige. • Hora establecida para el acceso a los locales de la entidad. • Verificar qué tipo de persona puede tener acceso. En el caso que sea visitante o empleado. • En el caso de empleados verificar quién es el responsable.
	Registrar salidas de Empleados y Visitantes.	• Hora establecida para la salida de los locales. • Verificar si el visitante salió a la hora que preciso por el responsable.
3	Listar Empleados y Visitantes que han tenido acceso a un local	• Mostrar los empleados que han tenido acceso a un local determinado. • Mostrar los visitantes que han tenido acceso a un local determinado.
	Listar locales a los que un Empleado y un Visitante han tenido acceso.	• Mostrar los locales a los que un empleado ha tenido acceso.

		• Mostrar los locales a los que un visitante ha tenido acceso.
	Mostrar listado de infracciones	• Listado de infracciones cometidas por un empleado en un rango de fecha determinado.

2.1.8 Arquitectura del sistema

Cliente-servidor

Para el desarrollo del módulo se propone la arquitectura cliente-servidor, que define la relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a la otra (servidor) y este último le remite las respuestas (López, 2010). Los principales componentes del módulo serían: un servidor local que brinda servicios con aplicaciones, varios clientes que realizan peticiones al servidor y una red que permite la conexión entre el servidor y el cliente.

Se hace uso del estilo arquitectónico en capas, el cual se basa en una distribución de los roles que interactúan directamente con el módulo y la gestión de locales donde los empleados tengan acceso.

Este estilo permite asignar correctamente las funcionalidades a cada capa, pudiéndose reutilizar las capas inferiores que no tengan dependencias con las superiores, evitando que los cambios en una de ellas afecte directamente al resto.

A continuación se muestra en la Figura 4 cómo está compuesta la arquitectura base para la organización estructural del módulo de control de acceso físico.

Figura 4. Arquitectura Cliente-Servidor.

2.1.9 Patrón de arquitectura

Modelo Vista Controlador (por sus siglas, MVC) es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación de la interfaz de usuarios con el módulo encargado de gestionar los eventos y las comunicaciones. Permite además separar cada una de las lógicas del módulo en archivos independientes permitiendo hacerlo mucho más flexible y sencillo de mantener (López, 2010).

Controlador: La principal función de esta capa es realizar una implementación de las funcionalidades definidas en las interfaces de la capa de negocio y al mismo tiempo trabajar directamente con la fuente de datos. Además es el componente que da soporte a las funcionalidades de la capa de negocio y se encuentra relacionada con la fuente de datos. El controlador que está interno en la plataforma, soporta el flujo de información del módulo de control de acceso físico, o sea la comunicación entre las vistas XML definidas y el módulo de control de acceso físico que está en el modelo.

Vista: Encapsula la presentación del módulo propuesto integrado en la plataforma, o sea que la vista no es más que la propia plataforma OpenERP. Las interfaces definidas muestran las funcionalidades para el registro y acceso de los empleados a los locales de la entidad, interactúa con la clase

controladora del OpenERP, la cual se encarga de procesar las interacciones del usuario, llamar a las clases del modelo y además realiza los cambios apropiados para la vista.

Modelo: El modelo contiene la información básica del módulo de control de acceso físico, métodos y clases que realizan el acceso y registro pertinente de los trabajadores. Esto incluye los datos y reglas de validación, así como acceso a datos y lógica de la programación.

En la Figura 5 se muestra la arquitectura Modelo-Vista-Controlador del módulo de control de acceso físico.

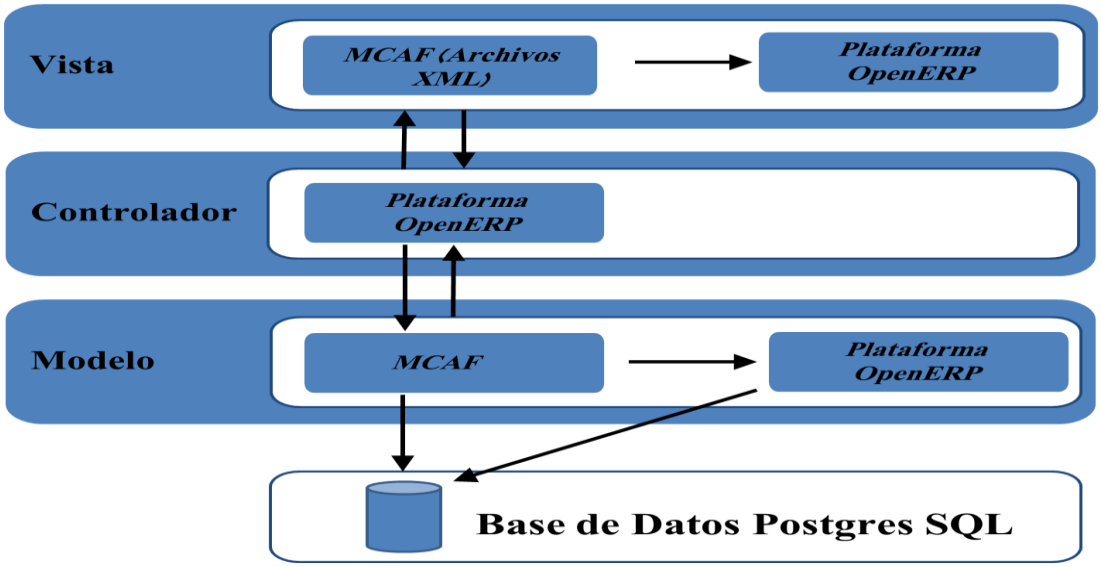


Figura 5. Arquitectura Modelo-Vista-Controlador.

El Modelo MVC es la representación específica de la información que maneja y gestiona el sistema. Para OpenERP, esa representación se hace mediante tablas en la base de datos en PostgreSQL. La Vista es la interfaz del usuario, es la representación del modelo expuesto de tal forma que la interacción del usuario con el sistema sea fácil, rápida e intuitiva. Tanto si el acceso se hace mediante la interfaz web, si el acceso es remoto, o mediante la aplicación cliente, si la instalación de OpenERP se ha hecho para cada PC, la Vista son los ficheros XML que presentan la información como listados, formularios, calendarios, gráficos, menús, informes, etc. Por último, el Controlador es quién da respuestas a las entradas generadas por el usuario, reacciona a los eventos que recibe el sistema. Dicho Controlador hace referencia al propio código python encargado de gestionar la información almacenada en el sistema, validando datos y realizando cálculos. (López, 2010).

2.1.10 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos relacionados entre sí, adaptado para resolver un problema de diseño general en un contexto particular. El mismo identifica: clases,

instancias, roles, colaboraciones y la distribución de responsabilidades. Estos modelos que se presentan como parejas de problema/solución con un nombre, codifican buenos principios y sugerencias relacionados con la asignación de responsabilidades, basados en la recopilación del conocimiento de los expertos en desarrollo de software (GAMMA, 2000).
En resumen los patrones de diseño se caracterizan por:

- Ser soluciones concretas.
- Ser soluciones técnicas.
- Se utilizan en situaciones frecuentes.
- Favorecen la reutilización de código.
- Su uso no se refleja en el código.
- Es difícil reutilizar la implementación de un patrón.

Los Patrones Generales de Software para Asignar Responsabilidades (por sus siglas en inglés, GRASP) describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, si se desea diseñar un software orientado a objetos con calidad. En el diseño de la aplicación propuesta se utilizaron los patrones GRASP experto, bajo acoplamiento y alta cohesión (Larman, 1999).

A continuación se analizará cada uno de los patrones de diseño utilizados en el desarrollo del módulo de control de acceso físico.

Patrón Experto

Consiste en asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Con este patrón se pretende que los objetos ejecuten acciones relacionadas con la información que poseen. Así se brinda soporte a una alta cohesión. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta también un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento (Larman, 1999).
En el módulo para el control del acceso físico este patrón se evidencia entre las clases AccesoEmpleado, AccesoVisitante y las clases Empleado, Visitante y Local, en el momento de monitorear el acceso a los locales de los empleados y visitantes, o sea el control del acceso , la lista de los accesos hechos por los usuarios que han tenido entrada y salida de los locales se muestra. Además se envidencia entre las clases Empleado, Visitante, Local y RegistraVisitante, RegistraEmpleado, donde la operación de registrar ya sea un empleado o un visitante emplea la asignación de responsabilidades desde las primeras clases.

Patrón Creador

Se basa en asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos y tiene como propósito fundamental encontrar un creador que se debe conectar con el objeto producido en cualquier evento (Larman, 1999). En el módulo este patrón se evidencia en las clases Empleado, Visitante y Local. Crean responsabilidades que las demás clases como RegistraEmpleado, RegistraVisitante, AccesoEmpleado, AccesoVisitante tienen que acceder a ellas para crear instancias de las mismas.

Patrón Alta Cohesión

Asigna una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que realicen un trabajo enorme. El patrón propone el diseño de clases con responsabilidades moderadas en su área funcional y que colabore con las otras para llevar a cabo una tarea (Larman, 1999).

En el diseño del módulo de control de acceso físico, como ejemplo de la aplicación de este patrón se encuentra la relación entre las clases RegistraEmpleado**,** RegistraSolapín**,** RegistraVisitante y Local, Empleado y Visitante, durante el proceso de registrar los usuarios empleados o visitantes en locales donde tengan el acceso.

Patrón Bajo Acoplamiento

Es una medida de la fuerza con que una clase está conectada a otras. El patrón propone el diseño de clases más independientes, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas (Larman, 1999).

Por ejemplo en el módulo del control del acceso físico se hace uso de este patrón en las clases RegistraEmpleado, RegistraSolapín**,** RegistraVisitante, AccesoEmpleado y AccesoVisitante durante el proceso de registro y acceso a los locales del personal autorizado, se reutiliza código en el atributo origen_id. Los beneficios que tiene el uso de este patrón es que las clases no se afectan por cambios de otras clases, son fáciles de entender por separado, además de ser fáciles de reutilizar.

2.1.11 Diagrama de clases del diseño

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos, puesto que muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Se utilizan para modelar la vista del diseño estática del software, especificar y documentar modelos estructurales, y para construir sistemas ejecutables, aplicando ingeniería directa e inversa (Larman, 1999).

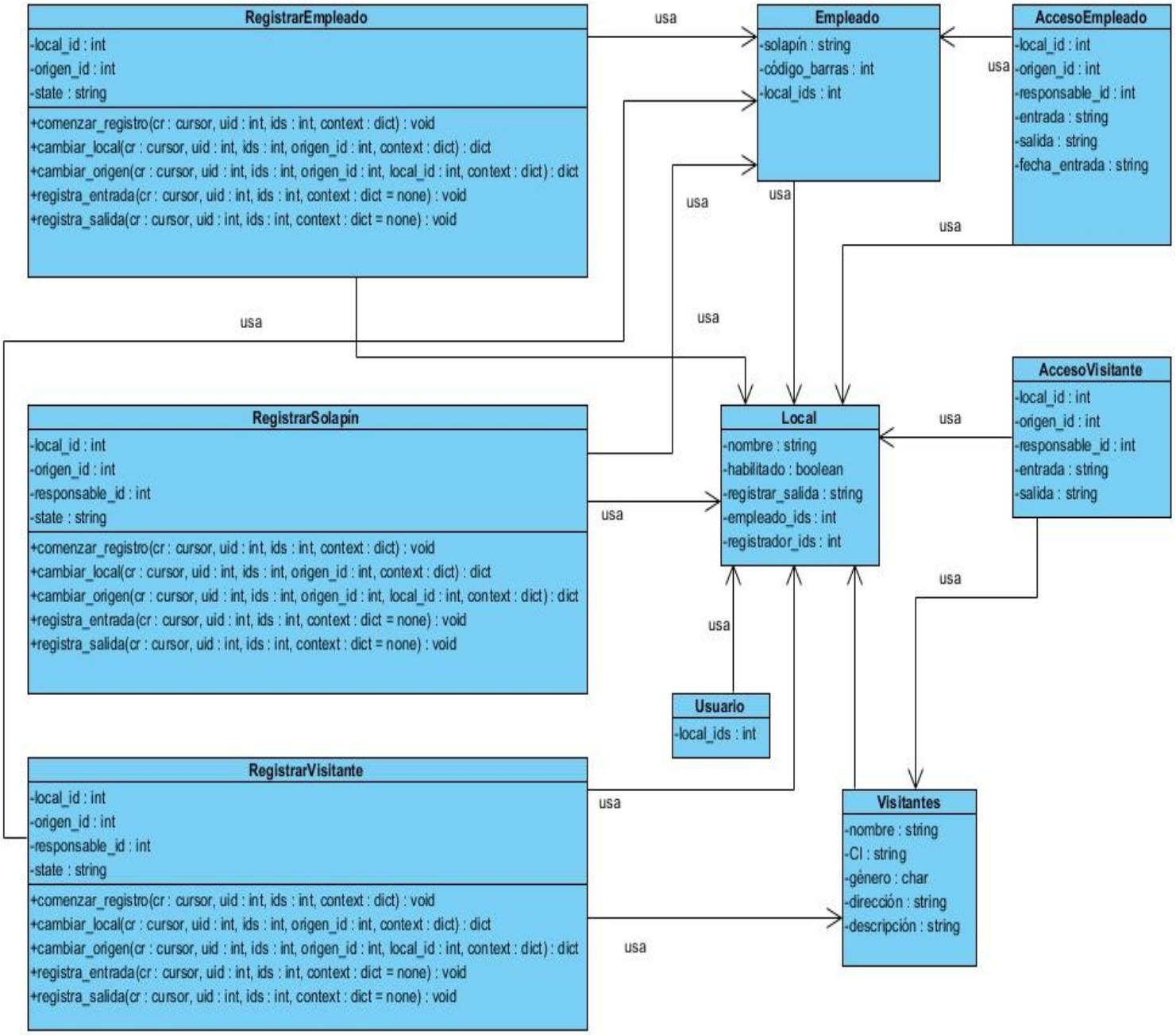


Figura 6. Diagrama de clases del diseño.

2.1.12 Tarjetas CRC

Las tarjetas CRC (Clase-Responsabilidad-Colaboración), constituyen uno de los artefactos de la metodología XP que guía el proceso de desarrollo de la solución propuesta. Se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase es cualquier persona, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las funcionalidades que realiza y sus atributos. Los colaboradores son las demás clases con las que interactúa con el fin de cumplir sus responsabilidades (Pressman, 2010). Seguidamente se exponen las tarjetas CRC correspondientes

a las clases que pertenecen a los dos primeros niveles especificados anteriormente, el resto se muestra en los anexos.

Tabla 6. Tarjeta CRC Clase AccesoEmpleado.

Clase AccesoEmpleado	
Responsabilidades	**Colaboradores**
Muestra el listado del registro de accesos de un empleado, entradas/salidas de un local determinado.	Empleado Local

Tabla 7. Tarjeta CRC Clase AccesoVisitante.

Clase AccesoVisitante	
Responsabilidades	**Colaboradores**
Muestra el listado del registro de accesos de un visitante, responsable del visitante, entradas/salidas de un local determinado.	Visitante Local Empleado

2.1.13 Conclusiones Parciales

La definición del modelo de dominio y la explicación detallada del proceso que debe ejecutar el módulo para el control del acceso físico permitieron identificar con mayor facilidad los requisitos funcionales que debe satisfacer la solución propuesta, propiciando así la comprensión de su funcionamiento.

Las historias de usuario especificadas para cada uno de estos requisitos favorecieron la estimación del plan de entregas del producto delimitando el propósito y duración de cada iteración.

Con el uso de la arquitectura cliente-servidor, el estilo arquitectónico MVC, y algunos de los patrones GRASP, se logró organizar la vista lógica de la solución, de manera que las clases identificadas y sus relaciones sean las bases para la implementación del módulo propuesto.

Capítulo 3 : Implementación y prueba de la solución.

Introducción

En este capítulo se exponen todas las particularidades de la implementación del módulo para llevar el control del acceso físico de personas, en aras de obtener un producto final que esté en correspondencia con los requisitos definidos, lo cual se validará aplicando diferentes tipos de prueba que propone la metodología de desarrollo de software utilizada. Para ello se definen los estándares de codificación que debe cumplir el equipo de desarrollo, se crea el diagrama de componentes y de despliegue, además de presentarse las interfaces de pruebas creadas para comprobar el funcionamiento de la solución que integrara a la plataforma. A partir del código resultante y su funcionamiento se ejecutan las pruebas unitarias, integración y de aceptación del módulo.

3.1 Conceptos fundamentales

A continuación se muestran características para la implementación y prueba del módulo del control del acceso físico.

3.1.1 Implementación

Una vez definidas las historias de usuario y concluido el diseño se pasa a la etapa de codificación de la solución propuesta cuyos objetivos van propuestos a desarrollar de forma iterativa e incremental un producto completo que esté preparado para la transición a su comunidad de usuarios, alcanzando versiones ventajosas de forma rápida y práctica, que paulatinamente completen la planeación, diseño, desarrollo y prueba de toda la funcionalidad necesaria.

La programación se realiza con el objetivo de crear código para cada historia de usuario en dependencia de lo pensado en el plan de iteraciones y las tareas ingenieriles. Alcanzando un mecanismo para la solución de problemas en tiempo real y también para el aseguramiento de la calidad en tiempo real. A medida que concluya cada iteración, el código desarrollado se integrará con el resto, puesto que esta estrategia de integración continua ayuda a evitar los problemas de compatibilidad e interfaces, y a descubrir a tiempo los errores.

3.1.1.1 Estándares de codificación.

En el proceso de desarrollo del módulo que propone la metodología XP se encuentran la refactorización del código y la propiedad compartida de éste. Para complementar estas prácticas, la metodología enfatiza que la comunicación de los programadores es a través del código, por lo cual es imprescindible que se sigan ciertos estándares de programación que proporcionen legibilidad. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único

programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Las técnicas de codificación añaden muchos aspectos al desarrollo del software, y aunque generalmente no afectan a la funcionalidad de la aplicación, sí contribuyen a una compresión efectiva del código fuente (Técnicas de codificación, 2014).

En la implementación del módulo de control del acceso físico se declaran los nombres de las variables, métodos y clases con las siguientes convenciones:

- En todos los casos se utilizarán nombres descriptivos que ayuden a una mejor comprensión del código. Ejemplo: Registrar_Visitante corresponde al registro de un visitante en local correspondiente.
- En los nombres, independiente del estilo que se utilice, las palabras van a estar separadas por un guión bajo.
- Para la nomenclatura de las clases y los métodos se utilizará el estilo de capitalización Pascal, con el cual se capitaliza la primera letra de cada palabra. Ejemplo: Registro_Empleado o Acceso_Visitante.
- No se utilizará una misma línea para definir más de una variable y siempre que sea posible éstas se inicializarán en su misma línea de declaración.
- Los nombres de las variables serán escritos completamente en minúsculas. Ejemplo: local_id, origen_id.

A continuación un ejemplo que la Figura 7 muestra del código con el estándar de codificación propuesto.

```
class accesofisico_acceso_empleado(osv.osv):
    _name = 'accesofisico.acceso.empleado'
    _description = 'Historial de acceso de empleados a locales.'
    _order = 'entrada desc'

    _columns = {
        'local_id': fields.many2one('accesofisico.locales', 'Local', required=True, readonly=True),
        'origen_id': fields.many2one('hr.employee', 'Empleado', required=True, readonly=True),
        'responsable_id': fields.many2one('hr.employee', 'Empleado', readonly=True),
        'entrada': fields.datetime('Entrada', required=True),
        'salida': fields.datetime('Salida', required=False, readonly=True),
        'fecha_entrada': fields.date('Fecha entrada', required=False, readonly=True, select=0),
    }

    _defaults = {
        'fecha_entrada': fields.date.context_today,
    }

    def accion_acceso_empleados(self, cr, uid, ids, domain=[], context=None):
        model_data = self.pool.get('ir.model.data')
        view_id = model_data.get_object_reference(cr, uid, 'accesofisico',
            'view_accesofisico_acceso_empleado_tree')[1]
        search_view_id = model_data.get_object_reference(cr, uid, 'accesofisico',
            'view_accesofisico_acceso_empleado_search')[1]
        ctx = dict(context or {})
        return {
            'type': 'ir.actions.act_window',
            'name': 'Acceso de empleados',
            'view_type': 'form',
            'view_mode': 'tree,form',
            'views': [(view_id, 'tree')],
            'res_model': 'accesofisico.acceso.empleado',
            'domain': domain,
            'view_id': view_id,
            'search_view_id': search_view_id,
            'context': ctx,
        }
```

Figura 7. Código de la clase de acceso de un empleado**.**

3.1.1.2 Diagrama de componentes.

Los diagramas de componentes modelan la vista estática del software, se representan como un grafo de componentes unidos por medio de relaciones de dependencia, mostrando las interfaces que soporten. Un componente es la parte modular de un sistema, desplegable y reemplazable, que encapsula la implementación, proporcionando la realización de un conjunto de interfaces. Típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, entre otros) (Larman, 1999). Son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas.

A continuación en la Figura 8 se muestra el diagrama definido para la solución propuesta.

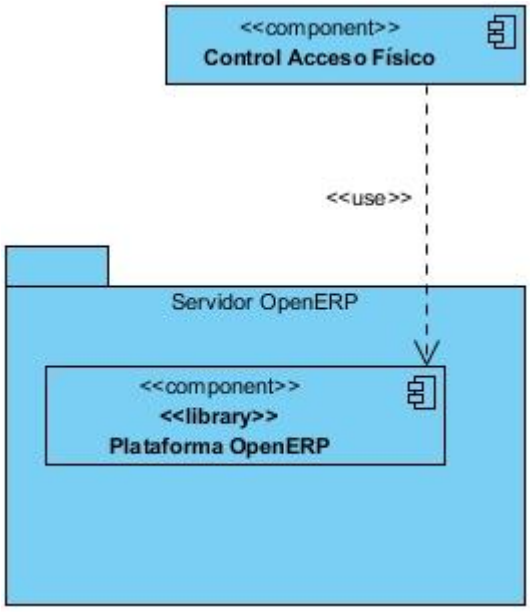


Figura 8. Diagrama de Componentes.

3.1.1.3 Diagrama de despliegue.

El modelo de despliegue es uno de los artefactos generados por la metodología usada durante el proceso de desarrollo del software, con el objetivo de capturar los elementos de configuración del procesamiento, todas las conexiones entre las características y visualizar su distribución. De manera general este tipo de modelo está compuesto por:

- Nodos o elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.
- Dispositivos caracterizados por ser nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- Conectores que expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo (Letelier, 2010).

El modelo de despliegue del módulo de control de acceso físico está compuesto solamente por tres nodos, que en este caso son: la PC-Cliente, el servidor de la plataforma OpenERP y el servidor de base de datos PostgresSQL. A continuación la Figura 9 muestra el diagrama de despliegue de la propuesta de solución.

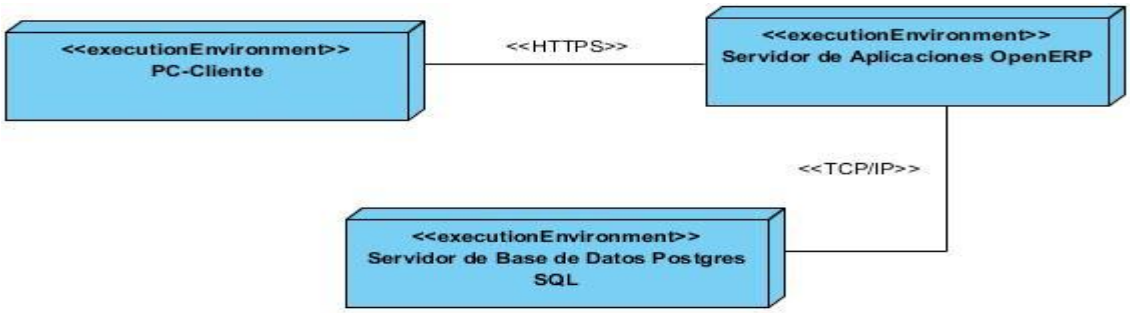


Figura 9. Diagrama de Despliegue.

3.1.1.4 Interfaces de usuario.

El objetivo del módulo creado es que tenga interfaces que visualicen todo el proceso de asignación y verificación del personal autorizado al acceso en los locales de una institución. No obstante, para comprobar y visualizar su correcto funcionamiento se define un conjunto de interfaces de pruebas.

A continuación se muestra la Figura 10 con la interfaz que indica el registro para el acceso de un empleado. Las interfaces restantes se muestran en los anexos (ver Anexo 1).

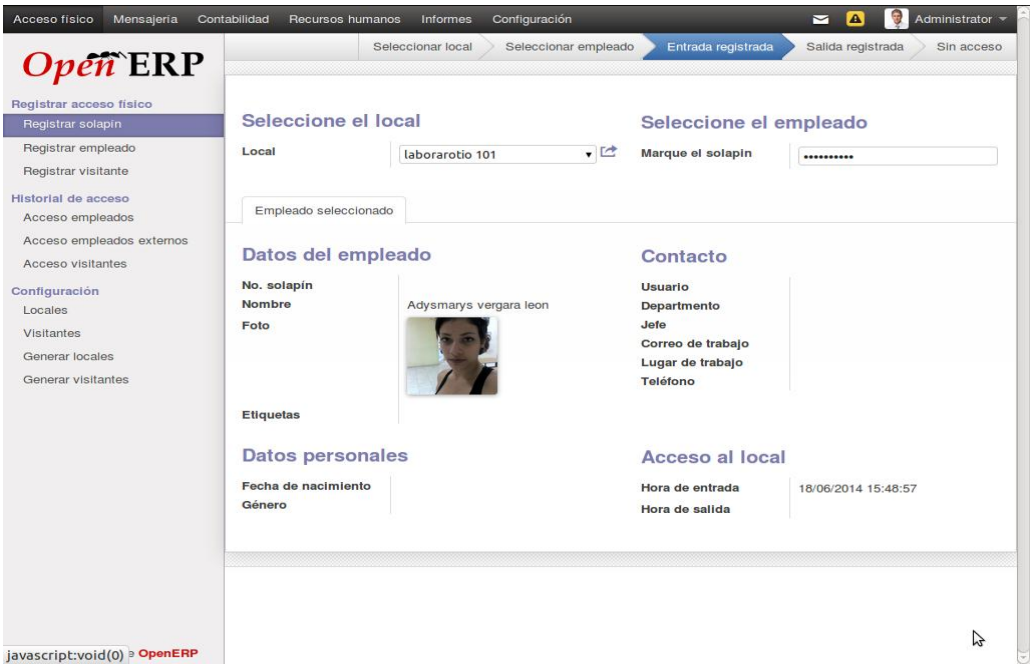


Figura 10. IU Registrar un empleado al sistema.

3.1.2 Pruebas

La prueba del software es un elemento crítico para el saneamiento de la calidad y representa una revisión final de las especificaciones del diseño y de la codificación. Además son actividades en las cuales un componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y se evalúa algún aspecto determinado del software (Parras, 2009).

Pruebas de Unitarias.

Se centran en el detallado, se probaron los caminos de control importantes con el fin de descubrir errores dentro del ámbito del módulo. La prueba de unidad hace uso intensivo de las técnicas de prueba de caja blanca (Parras, 2009).

Pruebas de Integración.

Son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Consiste en realizar pruebas para verificar que el módulo y la plataforma en su conjunto de partes de software funcionan juntos (Parras, 2009).

Pruebas de Aceptación.

Son básicamente pruebas funcionales, sobre el Sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario (Parras, 2009). Estas pruebas las tiene que realizar el cliente, pero por tratarse de un sistema implementado en CISED, el mismo se toma como cliente.

3.1.2.1 Pruebas Unitarias.

Están enfocadas a los elementos más pequeños del software. Son aplicables a funcionalidades para identificar y verificar que los flujos de control y de datos funcionan correctamente. La prueba de unidad siempre está orientada a caja blanca y el programador es el encargado de escribirlas, además de producir el código del sistema.

Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas, de modo que puedan ejecutarse en reiteradas ocasiones y con facilidad. Esto estimula una estrategia de pruebas de regresión, siempre que se modifique el código (Parras, 2009).

Para realizar las pruebas unitarias del módulo de control de acceso físico, se utilizó la librería “unittest”, que permite automatizar estas pruebas en python. En las siguientes tablas se muestran los casos de prueba aplicados a algunas de las funcionalidades más importantes de la propuesta de solución.

Tabla 8. Caso de Prueba Unitaria: test_vEntrada.

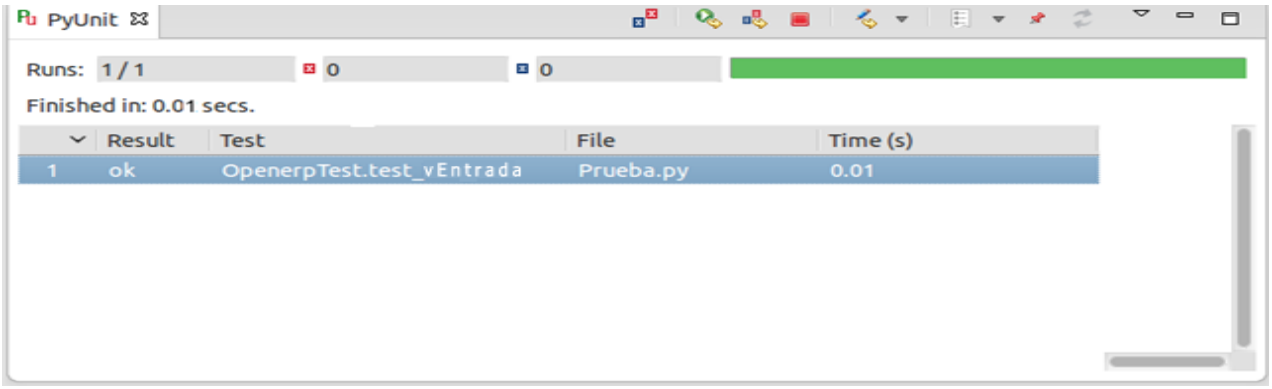
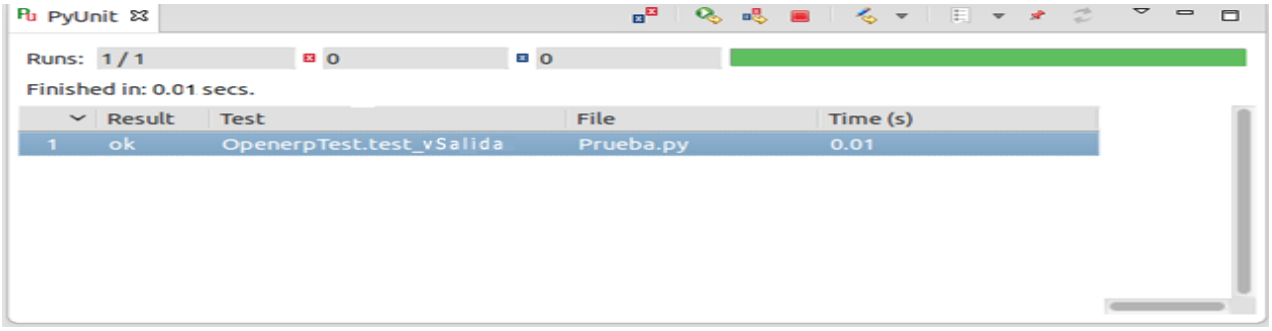
Caso de Prueba Unitaria	
Nombre de Prueba: test_vEntrada.	**ID:** 1
Descripción del caso de prueba: Responde al registro de entrada de un visitante.	
Responsable: Adysmarys Vergara León	
Criterio de Aceptación: Se realiza el registro de entrada a un visitante.	
Resultado: 	

Tabla 9. Caso de Prueba Unitaria: test_vSalida.

Caso de Prueba Unitaria	
Nombre de Prueba: test_vSalida.	**ID:** 1
Descripción del caso de prueba: Responde al registro de salida de un visitante.	
Responsable: Adysmarys Vergara León	
Criterio de Aceptación: Se realiza el registro de salida de un visitante.	
Resultado: 	

3.1.2.3 Pruebas de Integración

Son técnicas sistemáticas para construir la estructura del programa mientras al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados en unidad y estructurar un programa que esté de acuerdo con el que dicta el

diseño. La integración puede ser descendente si se integran los módulos desde el control o programa principal, o bien, ascendente, si la verificación del diseño empieza desde los módulos más bajos y de allí al principal. La selección de una estrategia de integración depende de las características del software y, a veces, del plan del proyecto, en algunos de los casos se puede combinar ambas estrategias (Pérez, 2012).

El módulo de control de acceso físico utiliza el enfoque ascendente para realizar las pruebas de integración, permitiendo construir y verificar las funcionalidades desde el nivel más bajo y finalizar en la funcionalidad principal, en este caso es el registro de entrada y salida de un empleado a los locales de la empresa.

A continuación se muestra el caso de prueba de integración correspondiente al proceso para el registro de entrada y salida de un empleado.

Tabla 10. Caso de Prueba de Integración: test_rEntrada.

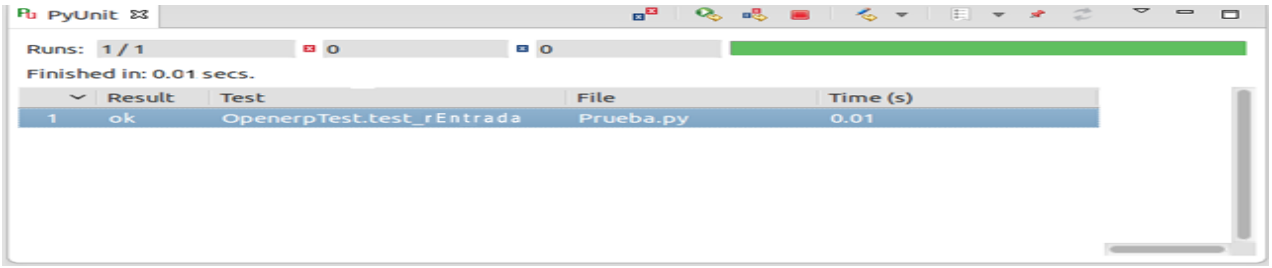
Caso de Prueba de Integración	
Nombre de Prueba: test_rEntrada.	**ID:** 1
Descripción del caso de prueba: Responde al registro de entrada de un empleado.	
Responsable: Adysmarys Vergara León	
Criterio de Aceptación: Se realiza el registro de entrada a un empleado	
Resultado: 	

Tabla 11. Caso de prueba de Integración: test_rSalida.

Caso de Prueba de Integración	
Nombre de Prueba: test_rSalida	**ID:** 2
Descripción del caso de prueba: Responde al registro de la salida de un empleado.	
Responsable: Adysmarys Vergara León	
Criterio de Aceptación: Se realiza el registro de salida de un empleado.	
Resultado:	

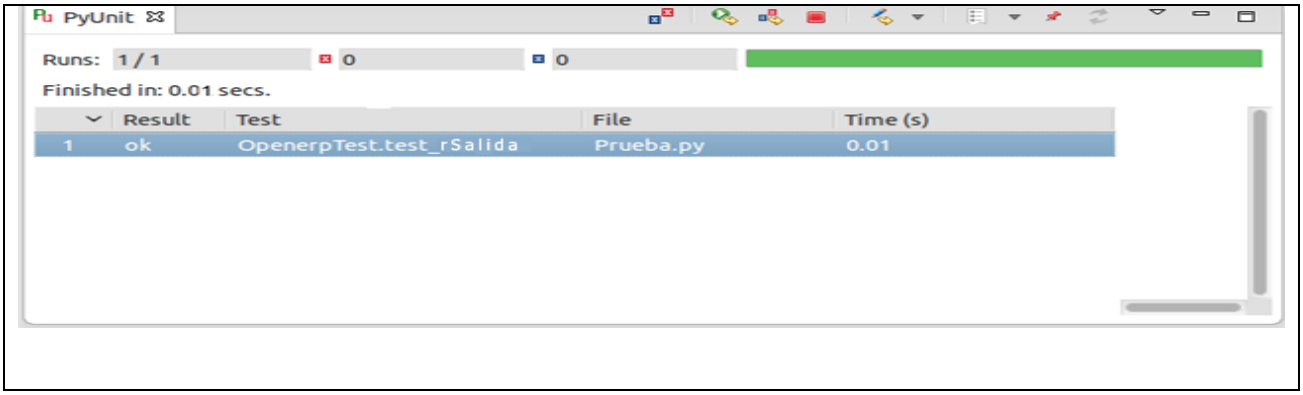
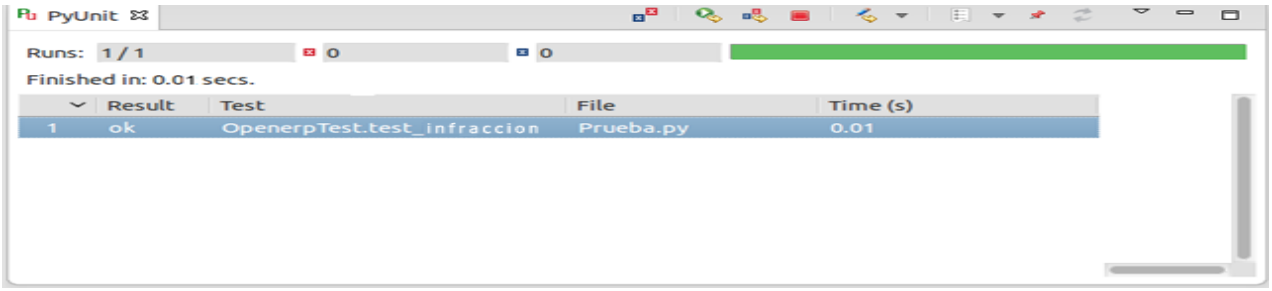


Tabla 12. Caso de Prueba de Integración: test_infraccion.

Caso de Prueba de Integración	
Nombre de Prueba: test_infraccion.	**ID:** 3
Descripción del caso de prueba: Identifica las infracciones cometidas por un empleado.	
Responsable: Adysmarys Vergara León	
Criterio de Aceptación: Se realiza el registro de entrada a un empleado	
Resultado: 	

3.1.2.4 Pruebas de Aceptación.

En el caso de XP, son creadas en base a las historias de usuario, en cada iteración del ciclo de desarrollo. La prueba de aceptación es la prueba final antes del despliegue del producto obtenido y su objetivo fundamental es verificar que el software esté listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones para las cuales fue construido.

Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que sus resultados sean correctos, y en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación (Joskowicz, 2008).

A continuación se muestra el caso de prueba de aceptación correspondiente a la historia de usuario “Gestionar locales en el sistema”. Los casos de prueba de aceptación definidos por el cliente para el resto de las historias de usuario se especifican en los anexos.

Tabla 13. Caso de Prueba de Aceptación: Gestionar locales en el sistema.

Caso de Prueba de aceptación	
Nombre de HU: Gestionar locales en el sistema.	**ID. de prueba: 1**
Nombre del caso de prueba: Gestionar locales a los cuales un usuario tenga acceso.	
Responsable: Adysmarys Vergara León	
Descripción: Permite crear locales a los cuales los usuarios tengan acceso.	
Condiciones de ejecución: El usuario debe ser un trabajador.	
Entrada/ Pasos de ejecución: Entrar en el módulo y acceder a ¨locales¨, donde se gestionan todos los locales, asignar los usuarios que pueden registrar su acceso en los mismos.	
Resultados Esperados: El sistema crea el local, dando la posibilidad de crean, eliminar o modificar.	
Evaluación de las pruebas: Prueba satisfactoria.	

3.1.2.5 Resultados de pruebas.

Los resultados de las pruebas fueron satisfactorios. Las pruebas unitarias permitieron identificar y verificar que los flujos de control y de datos funcionan correctamente: después de culminar el desarrollo del módulo de control de acceso físico se les realizó esta prueba a los métodos más importantes que lo conforman.

Con un total de 7 no conformidades respectivamente que revelaban errores en la codificación que en la mayoría de los casos no incidían significativamente en los resultados esperados, fueron solucionados en un corto plazo, antes de pasar a la siguiente iteración. En la Figura 11 se muestran la cantidad de no conformidades encontradas en cada iteración:

Figura 11. No conformidades detectadas en las 3 iteraciones.

Las pruebas de integración, donde su objetivo principal es detectar errores asociados con la interacción, solicitaron de pruebas unitarias en los métodos que responden al registro, monitoreo y acceso de empleados o visitantes en los locales de una empresa. Luego de resolver las no conformidades encontradas en las pruebas unitarias, se efectuaron las pruebas de integración a la funcionalidad "RegistraEmpleado" obteniendo una correcta integración al módulo de control de acceso físico.
Se realizaron además las pruebas de aceptación al módulo implementado con el objetivo de verificar que esté listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones para las cuales fue construido.

Las pruebas de aceptación, basadas en las historias de usuarios, se realizaron al finalizar la tercera iteración donde se logró obtener un módulo con la calidad requerida.

3.1.3 Conclusiones parciales.

En este capítulo fue presentada la etapa de implementación y prueba, donde el uso de los estándares de codificación definidos permitió desarrollar un código reutilizable haciendo más simple la puesta a punto de la solución propuesta. La representación del diagrama de componentes, el diagrama de despliegue y las interfaces de pruebas creadas, visualizaron la estructura y funcionamiento del módulo de control de acceso físico, en el cual tras la ejecución de las pruebas unitarias, integración y aceptación validaron el correcto funcionamiento en relación a los requisitos definidos para el desarrollo del módulo.

Conclusiones.

- Ninguna de las herramientas que realiza el control del acceso físico a nivel nacional e internacional puede integrarse a la plataforma OpenERP, debido a que no cumplen dos de los requisitos indispensables para ello, estar implementadas en python y utilizar el IDE de desarrollo OpenObject.
- Las herramientas y tecnologías seleccionadas permitieron desarrollar un módulo que al integrarse a la plataforma OpenERP permite realizar el control de acceso físico desde la misma, ello mediante el uso del lector de código de barras.
- Mediante la selección de una metodología ágil, se desarrolló la solución propuesta de una manera más dinámica y rápida, debido a que se dedicó más tiempo al desarrollo y no a la realización de artefactos propuestos en metodologías robustas.
- La realización de pruebas unitarias y de aceptación permitieron verificar el correcto funcionamiento del módulo propuesto, dándole así cumplimento a los requisitos funcionales identificados inicialmente.
- Al incorporar en la solución OpenERP un módulo encargado del control de acceso físico se logra automatizar el control de registro del personal que accede a las áreas o locales de las empresas o entidades donde se encuentre desplegada la plataforma. Permitiendo así un mejor control y monitoreo a la información asociada al personal que accede a un local determinado de dichas empresas. Demostrando además que la plataforma OpenERP es una solución integral que se ajusta a las necesidades crecientes de cualquier empresa o entidad.

Recomendaciones.

Aunque el uso del módulo integrado a la plataforma OpenERP permite llevar un control de acceso físico de las personas que tienen la previa autorización a la entrada/salida de una empresa, entidad o institución, se debe llevar a cabo un análisis de otras variantes que pueden ser utilizadas con el fin de proporcionar un mejoramiento del control y monitoreo de la seguridad. Por ello con el objetivo de seguir optimizando dicho proceso se recomienda que:

- Una vez que el módulo creado se integre a la plataforma, además de generar locales, empleados, realizar el registro y acceso a locales, se le incorporen nuevas funcionalidades para el monitoreo de entrada/salida de personas. Ejemplo agregar tecnología biométrica que proporcione más opciones en el control y monitoreo de la seguridad en una institución o entidad.

Bibliografía Consultada.

1 **Baeza, Pablo Nicolás. 2013.** Visual Paradigm DB Visual ARCHITECT SQL. *Visual Paradigm DB Visual ARCHITECT SQL.* [En línea] Enero de 2013. [Citado el: 10 de Enero de 2014.] http://www.docstoc.com/docs/96492173/Visual-Paradigm-Studio..

2 **Borrás, Jose Ramón Terol. 13 de septiembre del 2010.** *Implantación de OpenERP y Programación de un Conector con Básculas MAPAL.* Valencia : Escuela Técnica Superior de Ingeniería Informática, 13 de septiembre del 2010.

3 **David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli. 2007.** *Role-Based Access Control.* Estados Unidos de América : ARTECH HOUSE, 2007. ISBN 13: 978-1-59693-113-8.

4 **E. GAMMA, R. HELM, R. JOHNSON y J. VLISSIDES. 2000.** *Patrones de Diseño.* 2000.

5 **Eva Calvo López. 2010.** *Estudio y especificación de un problema de distribución a cliente final en una empresa tipo utilizando la herramienta OpenERP.* Terrassa(Barcelona) : Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII), 2010.

6 **Ibarra, Yamile Castro. 2012.** Cuba en la migración hacia el software libre. *Informática 2013.XV Convención y Feria Internacional Ministerio de la Informática y las Comunicaciones.* [En línea] XV Convención y Feria Internacional Ministerio de la Informática y las Comunicaciones., 2012. [Citado el: 14 de febrero de 2014.] http://www.informaticahabana.cu/node/320.

7 **Ivar Jacobson, Grady Booch, James Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000. ISBN-84-7829-036-2.

8 **Joskowicz, Ing. José. 2008.** *Reglas y Prácticas en eXtreme Programming.* España : Doctorado de Ingeniería Telemática de la Universidad, 2008.

9 **Larman, Craig. 1999.** *UML y Patrones.* 1999. *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos.* Mexico : PRENTICE HALL, 1999. ISBN: 970-17-0261-1.

10 *Metodologías de Desarrollo de Software.* **Sánchez, María A. Mendoza. Junio, 7 del 2004.** Perú : Perú S.A.C., Junio, 7 del 2004.

11 **Patricio Letelier, Mª Carmen Penadés. 2010.** *Metodologías ágiles para el desarrollo de software, Extreme Programming(XP).* España : Universidad Politpécnica de Valencia, 2010.

12 **Placio, Juan. 2006.** *El Modelo Scrum.* 2006. NST-0010/Rev. 0.1.

13 **Pressman, Roger S. 2010.** *Software Engineering: A Partitioner´s Approach.* s.l. : McGraw-Hill, 2010. ISBN 978-0-07-337-597-7.

14 **Roberth G. Figueroa, Camilo J. Solis.** *Metodologías tradicionales VS Metodologías ágiles.* s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

15 **Rumbaugh, Ivar Jacobson y James. 2000.** *El Lenguaje Unificado del Modelado.* s.l. : Manual de Referencia, 2000. ISBN: 8478290370 .

16 **Walter M. FLores Parras, Jose M. Luna , Nancy K. Maria Ochoa. 2009.** *SISTEMA INFORMATICO DE CONTROL EN ACTIVO FIJO Y TESORERIA.* SAN VICENTE : s.n., 2009.

17 **Wilson, Leslie Blackett. 1993.** *Comparative Programming Languages.* s.l. : Addison-Wesley, 1993. ISBN 0-201-56885-3.

Bibliografía Referenciada.

1 **Actum. 2014.** Software de Control de Acceso Biostar. *Actum.* [En línea] Actum, 2014. [Citado el: 29 de Octubre de 2013.] http://www.actum.es/soluciones-actum/control-de-acceso-biometrico/terminales-suprema/software-de-control-de-acceso-biostar.

2 **Baeza, Pablo Nicolás. 2013.** Visual Paradigm DB Visual ARCHITECT SQL. *Visual Paradigm DB Visual ARCHITECT SQL.* [En línea] Enero de 2013. [Citado el: 10 de Enero de 2014.] http://www.docstoc.com/docs/96492173/Visual-Paradigm-Studio..

3 **Borrás, Jose Ramón Terol. 13 de septiembre del 2010.** *Implantación de* OpenERP *y Programación de un Conector con Básculas MAPAL.* Valencia : Escuela Técnica Superior de Ingeniería Informática, 13 de septiembre del 2010.

4 **Cano, Jairo A. Gómez. 2014.** Sistema de Control de Acceso del Personal de la Empresa. *INBIOSYS Biometria.* [En línea] Blog de WordPress.com. El Tema K2-lite. RSS Entries and RSS Comments, 31 de enero de 2014. [Citado el: 14 de febrero de 2014.] http://inbiosys.wordpress.com/nuestros-productos/sistema-de-control-de-acceso-del-personal-de-la-empresa/.

5 **Colombia, Python. 2014.** Python Colombia. *Python.* [En línea] 2014. [Citado el: 9 de marzo de 2014.] https://sites.google.com/site/pythoncolombia.

6 **ConbotasSucias. 2012.** ConbotasSucias. *ConbotasSucias.* [En línea] 17 de Noviembre de 2012. [Citado el: 15 de Febrero de 2014.] http://conbotassucias.wordpress.com/2012/11/27/lector-de-codigos-de-barras/.

7 **DATYS Tecnología y Sistemas. 2011.** Control de Fronteras. *DATYS Tecnología y Sistemas.* [En línea] DATYS Tecnología y Sistemas, 2011. [Citado el: 4 de Noviembre de 2013.] http://www.datys.cu/wpinfproducto.aspx?50.

8 **—. 2011.** Sistema de Video Protección. *DATYS Tecnología y Sistemas.* [En línea] DATYS Tecnología y Sistemas, 2011. [Citado el: 29 de Octubre de 2013.] http://www.datys.cu/wpinfproducto.aspx?42.

9 **David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli. 2007.** *Role-Based Access Control.* Estados Unidos de América : ARTECH HOUSE, 2007. ISBN 13: 978-1-59693-113-8.

10 **Documentos. 2013.** es.lidocs.org. *es.lidocs.org.* [En línea] 2013. [Citado el: 5 de marzo de 2014.] http://es.lidocs.org/docs/index-232295.html.

11 **DokkoGroup. 2007-2014.** Making Software. *DokkoGroup.* [En línea] dokkogroup, 2007-2014. [Citado el: 14 de febrero de 2014.] http://www.dokkogroup.com.ar/.

12 **E. GAMMA, R. HELM, R. JOHNSON y J. VLISSIDES. 2000.** *Patrones de Diseño.* 2000.

13 **Eva Calvo López. 2010.** *Estudio y especificación de un problema de distribución a cliente final en una empresa tipo utilizando la herramienta* OpenERP*.* Terrassa(Barcelona) : Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII), 2010.

14 **Ibarra, Yamile Castro. 2012.** Cuba en la migración hacia el software libre. *Informática 2013.XV Convención y Feria Internacional Ministerio de la Informática y las Comunicaciones.* [En línea] XV Convención y Feria Internacional Ministerio de la Informática y las Comunicaciones., 2012. [Citado el: 14 de febrero de 2014.] http://www.informaticahabana.cu/node/320.

15 **Inc. Red Hat. 2005.** Manual de Seguridad. Capítulo1.Generalidades Sobre la Segguridad. *Red Hat Enterprise Linux.* [En línea] Red Hat,Inc., 2005. [Citado el: 22 de Octubre de 2013.] http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-es-4/s1-sgs-ov-controls.html.

16 **Ivar Jacobson, Grady Booch, James Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000. ISBN-84-7829-036-2.

17 **Joskowicz, Ing. José. 2008.** *Reglas y Prácticas en eXtreme Programming.* España : Doctorado de Ingeniería Telemática de la Universidad, 2008.

18 **Kordauz. 2007-2014.** Control Acceso Físico. *Technology Integrator.* [En línea] Tecnológística de Costa Rica, 2007-2014. [Citado el: 22 de Octubre de 2013.] http://www.grupotec.com/index.php?page=control-de-acceso&hl=esp.

19 **Larman, Craig. 1999.** *UML y Patrones.* 1999.

20 **—. 1999.** *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos.* Mexico : PRENTICE HALL, 1999. ISBN: 970-17-0261-1.

21 **Loaiza, Douglas Alfredo. 2012.** Metodología XP. *Ingeniería de Software.* [En línea] 12 de Julio de 2012. [Citado el: 25 de Noviembre de 2013.] http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html.

22 **Martinez, Rafael. 1998-2012.** Sobre Linux. *El rincón de Linux .* [En línea] El rincón de Linux, 1998-2012. [Citado el: 15 de marzo de 2014.] http://www.linux-es.org/sobre_linux.

23 *Metodologías de Desarrollo de Software.* **Sánchez, María A. Mendoza. Junio, 7 del 2004.** Perú : Perú S.A.C., Junio, 7 del 2004.

24 **Microsoft. 2014.** Técnicas de codificación. *Developed Network.* [En línea] 2014. [Citado el: 8 de abril de 2014.] http://msdn.microsoft.com/es-es/library/aa291593%28v=vs.71%29.aspx.

25 **MIKE. 2012.** Las pruebas de integración de software. [En línea] Telemex, 25 de septiembre de 2012. [Citado el: 16 de abril de 2014.] http://www.academica.mx/blogs/las-pruebas-integraci%C3%B3n-software.

26 **Patricio Letelier, Mª Carmen Penadés. 2010.** *Metodologías ágiles para el desarrollo de software, Extreme Programming(XP).* España : Universidad Politpécnica de Valencia, 2010.

27 **Placio, Juan. 2006.** *El Modelo Scrum.* 2006. NST-0010/Rev. 0.1.

28 **Pressman, Roger S. 2010.** *Software Engineering: A Partitioner´s Approach.* s.l. : McGraw-Hill, 2010. ISBN 978-0-07-337-597-7.

29 **Rafael Martinez.** [En línea] [Citado el: 29 de Enero de 2014.] http://www.postgresql.org.es/sobre_postgresql.

30 **Roberth G. Figueroa, Camilo J. Solis.** *Metodologías tradicionales VS Metodologías ágiles.* s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

31 **Rumbaugh, Ivar Jacobson y James. 2000.** *El Lenguaje Unificado del Modelado.* s.l. : Manual de Referencia, 2000. ISBN: 8478290370 .

32 **Sistemas, DATYS Tecnología y. 2011.** http://www.datys.cu/wpinfproducto.aspx?42. *DATYS Tecnología y Sistemas.* [En línea] DATYS Tecnología y Sistemas, 2011. [Citado el: 29 de noviembre de 2013.] http://www.datys.cu/wpinfproducto.aspx?42.

33 **Sistemas, Terminales y Software para el Control de Accesos, Horario y Presencia. 2010.** CS-Access. Software de Control de Acceso para PYMES. *Grupo SPEC.* [En línea] Sistemas, Terminales y Software para el Control de Accesos, Horario y Presencia, 2010. [Citado el: 22 de Octubre de 2013.] http://www.grupospec.com/productos/singular-tech/cs-access-software-de-control-de-acceso-para-pymes.

34 **Subinet. 2010.** Software de Control de Acceso AMADEUS 5. *Subinet.* [En línea] Subinet, 14 de Septiembre de 2010. [Citado el: 29 de Octubre de 2013.] http://www.subinet.es/guias-y-tips/guias-y-tips-seguridad/%C2%BFcuales-son-los-modelos-de-control-de-.

35 **Visual Paradigm. 2013.** Visual Paradigm. *Visual Paradigm.* [En línea] enero de 2013. [Citado el: 10 de enero de 2014.] http://www.visual-paradigm.com/.

36 **Wilson, Leslie Blackett. 1993.** *Comparative Programming Languages.* s.l. : Addison-Wesley, 1993. ISBN 0-201-56885-3.

37 **Windows. 2013.** Historia de Windows. *Windows.* [En línea] Windows, noviembre de 2013. [Citado el: 15 de febrero de 2014.] http://windows.microsoft.com/es-419/windows/history#T1=era0.

38 **1993***Comparative Programming Languages*Addison-Wesley1993ISBN 0-201-56885-3

39 **2012**ConbotasSucias*ConbotasSucias*http://conbotassucias.wordpress.com/2012/11/27/lector-de-codigos-de-barras/

40 **2010**CS-Access. Software de Control de Acceso para PYMES*Grupo SPEC*Sistemas, Terminales y Software para el Control de Accesos, Horario y Presenciahttp://www.grupospec.com/productos/singular-tech/cs-access-software-de-control-de-acceso-para-pymes

41 **2012**Cuba en la migración hacia el software libre*Informática 2013.XV Convención y Feria Internacional Ministerio de la Informática y las Comunicaciones.*XV Convención y Feria Internacional Ministerio de la Informática y las Comunicaciones.http://www.informaticahabana.cu/node/320

42 **DATYS Tecnología y Sistemas2011**Control de Fronteras*DATYS Tecnología y Sistemas*DATYS Tecnología y Sistemashttp://www.datys.cu/wpinfproducto.aspx?50

43 **2011**Sistema de Video Protección*DATYS Tecnología y Sistemas*DATYS Tecnología y Sistemashttp://www.datys.cu/wpinfproducto.aspx?42

44 **Documentos2013**es.lidocs.org*es.lidocs.org*http://es.lidocs.org/docs/index-232295.html

45 **2000***El Lenguaje Unificado del Modelado*Manual de Referencia2000ISBN: 8478290370

46 **2006***El Modelo Scrum*2006NST-0010/Rev. 0.1

47 **2000***El Proceso Unificado de Desarrollo de Software*Addison Wesley2000ISBN-84-7829-036-2

48 **Eva Calvo López2010***Estudio y especificación de un problema de distribución a cliente final en una empresa tipo utilizando la herramienta* OpenERP.Terrassa(Barcelona)Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII)2010

49 **2013**Historia de Windows*Windows*Windowshttp://windows.microsoft.com/es-419/windows/history#T1=era0

50 **2011**http://www.datys.cu/wpinfproducto.aspx?42*DATYS Tecnología y Sistemas*DATYS Tecnología y Sistemashttp://www.datys.cu/wpinfproducto.aspx?42

51 **13 de septiembre del 2010***Implantación de* OpenERP *y Programación de un Conector con Básculas MAPAL*ValenciaEscuela Técnica Superior de Ingeniería Informática13 de septiembre del 2010

52 **Inc. Red Hat2005**Manual de Seguridad. Capítulo1.Generalidades Sobre la Segguridad*Red Hat Enterprise Linux*Red Hat,Inc.http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-es-4/s1-sgs-ov-controls.html

53 **Kordauz2007-2014**Control Acceso Físico*Technology Integrator*Tecnológística de Costa Ricahttp://www.grupotec.com/index.php?page=control-de-acceso&hl=esp

54 **2012**Las pruebas de integración de softwareTelemexhttp://www.academica.mx/blogs/las-pruebas-integraci%C3%B3n-software

55 **2007-2014**Making Software*DokkoGroup*dokkogrouphttp://www.dokkogroup.com.ar/

56 **2012**Metodología XP*Ingeniería de Software*http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html

57 **2010***Metodologías ágiles para el desarrollo de software, Extreme Programming(XP)*EspañaUniversidad Politpécnica de Valencia2010

58 *Metodologías de Desarrollo de Software* **Junio, 7 del 2004**PerúPerú S.A.C.

59 *Metodologías tradicionales VS Metodologías ágiles*Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación

60 **Microsoft2014** Técnicas de codificación*Developed Network*http://msdn.microsoft.com/es-es/library/aa291593%28v=vs.71%29.aspx

61 **2000** *Patrones de Diseño*2000

62 **2014** Python Colombia*Python*https://sites.google.com/site/pythoncolombia

63 **Rafael Martinez**http://www.postgresql.org.es/sobre_postgresql

64 **2008** *Reglas y Prácticas en eXtreme Programming*EspañaDoctorado de Ingeniería Telemática de la Universidad2008

65 **2007** *Role-Based Access Control*Estados Unidos de AméricaARTECH HOUSE2007ISBN 13: 978-1-59693-113-8

66 **2014** Sistema de Control de Acceso del Personal de la Empresa*INBIOSYS Biometria*Blog de WordPress.com. El Tema K2-lite. RSS Entries and RSS Commentshttp://inbiosys.wordpress.com/nuestros-productos/sistema-de-control-de-acceso-del-personal-de-la-empresa/

67 **1998-2012** Sobre Linux *El rincón de Linux* El rincón de Linuxhttp://www.linux-es.org/sobre_linux

68 **2010** Software de Control de Acceso AMADEUS 5*Subinet*Subinethttp://www.subinet.es/guias-y-tips/guias-y-tips-seguridad/%C2%BFcuales-son-los-modelos-de-control-de-

69 **2014**Software de Control de Acceso Biostar*Actum*Actumhttp://www.actum.es/soluciones-actum/control-de-acceso-biometrico/terminales-suprema/software-de-control-de-acceso-biostar

70 **2010***Software Engineering: A Partitioner´s Approach*McGraw-Hill2010ISBN 978-0-07-337-597-7

71 **1999** *UML y Patrones*1999

72 **1999** *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos.*MexicoPRENTICE HALL1999ISBN: 970-17-0261-1

73 **2013** Visual Paradigm DB Visual ARCHITECT SQL*Visual Paradigm DB Visual ARCHITECT SQL*http://www.docstoc.com/docs/96492173/Visual-Paradigm-Studio.

74 **Visual Paradigm2013** Visual Paradigm*Visual Paradigm*http://www.visual-paradigm.com/

Glosario de términos.

A

Acceso: es el resultado positivo de una autentificación.

C

CISED: Centro de Identificación y Seguridad Digital.

Controles de seguridad: definen los objetivos principales de una seguridad apropiada.

Control de acceso físico: Mediante un manejo avanzado de credencialización que permite controlar, limitar, monitorear y auditar el acceso físico. Este tipo de sistema es ideal para organizaciones que desea controlar una única área restringida o múltiples puertas de acceso.

E

ERP: Planificador de Recursos Empresariales.

Estándar: normas que ofrecen un lenguaje de punto común de comunicación.

F

Flujos de Trabajo: estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

G

Gestión Empresarial: actividad empresarial que a través de diferentes individuos buscarán mejorar la productividad y la competitividad de una empresa o de un negocio.

I

IDE: Entorno Integrado de Desarrollo.

O

OpenERP*:* Sistema de Gestión Empresarial.

P

Plataforma OpenERP*:* Es un completo sistema de gestión empresarial que cubre las necesidades de muchas áreas, ejemplo contabilidad, finanzas, ventas, RRHH, compras, proyectos y almacén.

R

RRHH**:** Módulo de Recursos Humanos que está integrado en la plataforma OpenERP, se incluye en los procesos del módulo para el control del acceso físico.

T

TIC**:** Tecnologías de la Informática y las Comunicaciones.

U

UCI**:** Universidad de las Ciencias Informáticas.