2018

Universidad de las Ciencias Informáticas

Facultad 1

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

Módulo de Control de Acceso para el Sistema de Información Primaria de Personas

Autor: Christian Marlon Paneque Moreda

Tutor(es): Ing. Mayleidis López Fernández

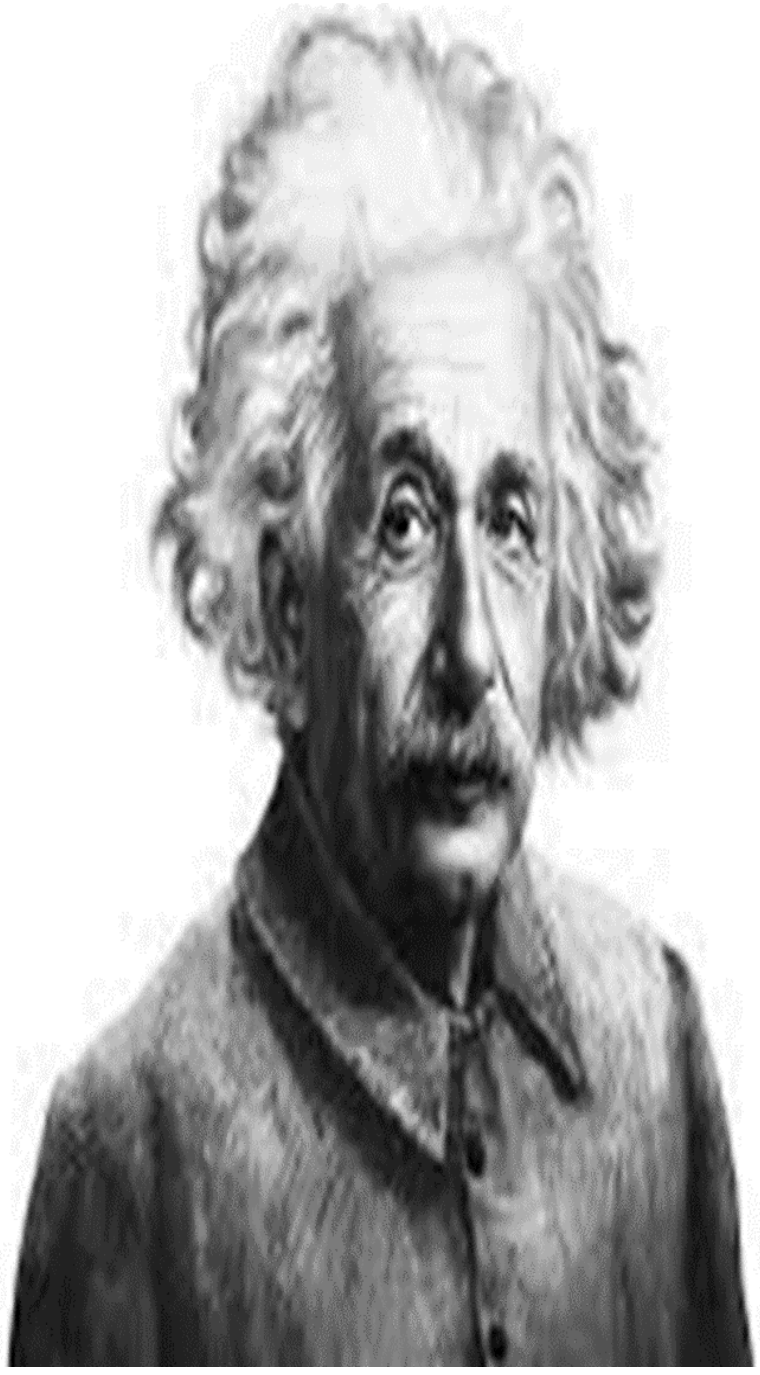
Ing. Osay Gonzáles Fuentes

La Habana Cuba

"Año 60 de la Revolución"

"La imaginación es más importante que el conocimiento. El conocimiento es limitado, mientras que la imaginación no "

Albert Einstein



Agradecimientos

Le agradezco a mi familia por estar a mi lado en todas mis batallas.

A Yasser y Reynaldo, por su incondicionalidad, y por estar presentes siempre que los necesite.

A Carlos, Evelyn, y Carli por, más que mis suegros y cuñado, haber sido mi segunda familia.

A mis amigos, en especial a mis Carlos, por tener que soportarme y ser mis hermanos en la vida.

A Mayleidis y Osay, por haber sido mi guía y mi ejemplo a seguir, por su humildad y apoyo.

A mis compañeros del 111 102, los llevaré siempre conmigo.

A todas esas personas que de una forma u otra contribuyeron a mi formación humana y científica y que no puedo mencionar porque se me acaba la hoja.

Dedicatoria

A las cinco puntas de mi estrella, Iraida, Suradma, Sahily, Daniela, y Evelyn, cinco mujeres como gigantes, ustedes son y serán mi luz.

Declaración de autoría

Declaro por este medio que yo, Christian Marlon Paneque Moreda, con carnet de identidad 93022434282 soy el autor principal del trabajo titulado “Módulo de Control de Acceso para el Sistema de Información Primaria de Personas (SIPP)”, y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos el presente documento a los __ días del mes de _____ del año 2018.

Christian M. Paneque Moreda

_____

Firma del Autor

Mayleidis López Fernández

_____

Firma del Tutor

Osay Gonzáles Fuentes

_____

Firma del Tutor

Resumen

Con el avance del desarrollo de la informática y las comunicaciones a nivel mundial se ha evidenciado la necesidad de controlar el acceso a los grandes volúmenes de información y recursos generados por las nuevas tecnologías. Cuba, se encuentra impulsada por la necesidad de actualizar y optimizar sus procesos, siendo una herramienta de apoyo la implementación de soluciones informáticas que satisfagan estas necesidades. El Sistema de Información Primaria de Personas (SIPP) constituye una de estas soluciones, desarrollada por el Centro de Identificación y Seguridad Digital (CISED); el SIPP fomentará la integración de los sistemas desarrollados por la Universidad de las Ciencias Informáticas (UCI) eliminando la duplicidad en la información y los potenciales efectos que esto puede ocasionar. La presente investigación se enfoca en la implementación de un módulo de control de acceso para el SIPP que permita realizar los procesos de control de acceso a las distintas áreas de una entidad. Para lograr el desarrollo de la solución se realizó un análisis de sistemas homólogos de control de acceso a nivel nacional e internacional, quedando definida AUP-UCI como metodología de desarrollo de la investigación, así como, las herramientas que serán utilizadas para la implementación. La implementación del Módulo de Control de Acceso y Registro de Visitantes para el SIPP permitirá garantizar la seguridad de la información y conocer el flujo de acceso a las distintas áreas de una entidad.

Palabras clave: controlar, acceso, módulo, seguridad

Summary

With the advance of the development of information technology and communications worldwide, the need to control access to the large volumes of information and resources generated by new technologies has become evident. Cuba is driven by the need to update and optimize its processes, being a tool to support the implementation of IT solutions that meet these needs. The Primary Persons Information System (SIPP) is one of these solutions, developed by the Digital Identification and Security Center (CISED); The SIPP will promote the integration of the systems developed by the University of Information Sciences (UCI), eliminating the duplication of information and the potential effects that this may cause. The present investigation focuses on the implementation of an access control module for the SIPP that allows the access control processes to be carried out in the different areas of an entity. To achieve the development of the solution, an analysis of homologous systems of access control at national and international level was carried out, with AUP-UCI being defined as a research development methodology, as well as the tools that will be used for implementation. The implementation of the Access Control and Visitor Registration Module for the SIPP will guarantee the security of the information and know the flow of access to the different areas of an entity.

Keywords: to control, access, module, security

Índice

Índice de tablas:

Índice de ilustraciones:

Introducción

La aplicación de las Tecnologías de la Información y las Comunicaciones (TIC) con el objetivo de optimizar complejos procesos en los distintos campos de dominio de la ciencia es un tema recurrente en nuestra actualidad. El uso de herramientas, técnicas y métodos para la realización de determinadas tareas en las esferas de la sociedad actual han evolucionado adaptándose a la utilización de las TIC. Uno de los sectores que evidencian la importancia del uso de las mismas como método de perfeccionamiento e innovación en sus procesos, tanto de negocios como administrativos, es el empresarial, en este ámbito la aplicación de soluciones informáticas ha demostrado ser imprescindible para la evolución tanto de la toma de decisiones administrativas como en la automatización de procesos y la seguridad de la información.

Cuba en los últimos años, ha dado sus primeros pasos en el desarrollo de aplicaciones informáticas en respuesta a la creciente necesidad de controlar sus recursos empresariales y fomentar la competitividad a nivel internacional. La informatización de la sociedad se enfoca en el proceso de utilización ordenada y masiva de las tecnologías de la informática y las comunicaciones para satisfacer las necesidades de información y conocimiento de la sociedad y como parte fundamental de este proceso se crea la Universidad de Ciencias Informáticas (UCI).

La UCI es una parte esencial en el desarrollo tecnológico de nuestro país con la implementación e instalación de más de un centenar de aplicaciones informáticas en los ministerios y sectores empresariales, tanto de nuestro país como a nivel internacional. Actualmente, estos sistemas producidos y comercializados por la universidad no intercambian entre ellos la información de las personas de una misma empresa; lo que conlleva a la duplicidad de datos en múltiples sistemas aumentando las probabilidades de error en información nominal de las personas. La duplicidad de información aumenta de manera considerable la necesidad de hardware, específicamente el almacenamiento y el procesamiento de los datos. Casi todos los subsistemas que componen el ecosistema de software de una entidad contienen datos nominales de personas que al ser expuestos para su consumo por otras aplicaciones comprometen la seguridad, integridad y confidencialidad de los datos.

El sistema único de gestión de personas actualmente en desarrollo en la Universidad de las Ciencias Informáticas no tiene un sistema de control de acceso y registro de visitantes, por lo que no garantiza el control de las personas que visitan y acceden diariamente a la entidad donde sea desplegado. El sistema cuenta con un conjunto de módulos que facilitan el control de los datos de las personas, las reglas de acceso

y las configuraciones de los puntos de acceso, así como los tipos de acceso sin embargo estas potencialidades no son utilizadas para el control de los visitantes y del personal, lo que conlleva a realizar estos procesos de manera separada.

Para eliminar este tipo de incidente el Centro de Identificación y Seguridad Digital (CISED) se encuentra elaborando una solución donde a partir de la captura de los datos primarios de una persona, brinde a los sistemas que lo necesiten esta información, eliminando así tanto la duplicidad de información y permitiendo controlar el acceso a los recursos, gestionar usuarios y sus datos de identificación, asociar roles, perfiles y políticas de seguridad para cada aplicación o producto de la universidad.

El Sistema de Información Primaria de Personas (SIPP), permitirá la eliminación de los problemas planteados anteriormente, y tendrá la capacidad de ser conformado como un producto genérico para ser implantado en otras empresas del país. Este sistema deberá ser capaz de mantener un estricto control de la información y recursos físicos en su entorno de despliegue, brindando la capacidad de conocer y controlar el flujo de acceso de los trabajadores y visitantes de las distintas áreas de una entidad.

Luego de analizar las distintas problemáticas mencionadas anteriormente se define el siguiente **problema de investigación**: ¿Cómo lograr la seguridad de la información y el control del flujo de acceso para el SIPP?

Definiéndose como **objeto de estudio**: Los procesos para el control de acceso.

Para dar solución al problema antes expuesto se plantea el siguiente **objetivo general:** Desarrollar un módulo de control de acceso y registro de visitantes para el SIPP que logre la seguridad de la información y el control del flujo de acceso a las distintas áreas de la entidad.

Para dar cumplimiento al objetivo general se definen como **objetivos específicos:**

- Analizar los antecedentes y bases de los sistemas dedicados al control de acceso físico y lógico, así como la base conceptual necesaria para describir el proceso.
- Realizar un análisis de la estructura básica del sistema empresarial cubano.
- Seleccionar la base tecnológica necesaria para la implementación del sistema de control de acceso.
- Diseñar el módulo para el control de acceso.
- Implementar el módulo para el control de acceso.

- Verificar la calidad del módulo de control de acceso para el SIPP.

Hipótesis de investigación: La implementación del módulo de control de acceso y registro de visitantes para el SIPP permitirá lograr la seguridad de la información y el control del flujo de acceso a las distintas áreas de la entidad.

Durante la realización de la presente investigación se utilizaron diversos métodos y procedimientos teóricos y empíricos, a continuación, se mencionan algunos de ellos:

Métodos teóricos:

- **Histórico-Lógico:** con el objetivo de estudiar la evolución de las características de los controladores de acceso a nivel mundial, con el fin de apoyar el proceso de modelado.
- **Hipotético-Deductivo:** para una mejor comprensión del negocio, y la definición de la hipótesis de investigación.

Métodos empíricos:

- **Observación:** con el fin de recopilar datos de procesos similar en nuestro entorno, así como estudiar los procesos de desarrollo de nuestra solución.
- **Experimental:** utilizado para comprobar la utilidad de del producto obtenido mediante la introducción de datos ficticios.

El documento se encuentra estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica, se realiza con el objetivo de estudiar y analizar los aspectos teóricos relacionados con el control de acceso en sistemas informáticos. Además, se despliegan y adoptan los principales conceptos que facilitaran el estudio y compresión del control de acceso y seguridad de la información.

Capítulo 2: Análisis y diseño del módulo de control de acceso y registro de visitantes, se lleva a cabo la modelación del negocio con el fin de entender el contexto del módulo a desarrollar, se recogen los requerimientos funcionales y no funcionales que debe cumplir el mismo. Se realiza la descripción de la solución propuesta, la arquitectura, interfaces y entidades necesarias para la implementación del software

Capítulo 3: Implementación y pruebas al módulo, incluye los distintos componentes que conforman al producto, se modela el despliegue del sistema y las interfaces de usuario, se describen los estándares de codificación que se tuvieron en cuenta en la implementación de la propuesta de solución, se presentan los tipos de prueba realizados para la verificación de la calidad del software y los resultados obtenidos

Finalmente se presentan las conclusiones y recomendaciones de la investigación.

Capítulo 1: Fundamentación teórica

En este capítulo serán expuestos y analizados los principales conceptos asociados al problema planteado, así como el estado en lo que a teoría y práctica se refiere sobre el tema de estudio. Se hará referencia a elementos de la investigación y sus antecedentes, así como los principales sistemas homólogos actuales. Serán descritas las tecnologías, herramientas y metodologías a utilizar en la solución propuesta.

Conceptos asociados al dominio del problema

Entre los factores indispensables para alcanzar el éxito en el desarrollo de aplicaciones e investigaciones que ofrezcan soluciones óptimas e integrales, se encuentra la calidad y claridad de las bases conceptuales del dominio del problema. Con este objetivo se definen a continuación los principales conceptos tratados en la investigación.

Sistema de Información:

Partiendo del marco de desarrollo de la investigación y la constatación de la existencia de múltiples definiciones del concepto de sistema de información(SI), se asume el concepto propuesto por el Ministerio de Justicia de Cuba en el Decreto -Ley No. 281.El cual expresa que “un Sistema de Información es un conjunto organizado de personas, procesos y recursos, incluyendo la información y sus tecnologías asociadas, que interactúan de forma dinámica, para satisfacer las necesidades informativas que posibilitan alcanzar los objetivos de una o varias organizaciones” . Se asumirá esta definición por el carácter genérico de la misma, así como la válida relación que presenta entre los SI, usuarios, organizaciones, procesos y tecnologías, conceptos fundamentales en la presente investigación.

Seguridad en Sistemas de Información

La información es el principal activo de muchas organizaciones y precisa ser protegida adecuadamente frente a amenazas que puedan poner en peligro la continuidad del negocio. En la actualidad, las empresas de cualquier tipo o sector de actividad se enfrentan cada vez más con riesgos e inseguridades procedentes de una amplia variedad de contingencias, las cuales pueden dañar considerablemente tanto los sistemas de información como la información procesada y almacenada.

Ante estas circunstancias, las organizaciones han de establecer estrategias y controles adecuados que garanticen una gestión segura de los procesos del negocio, priorizando la protección de la información. Para proteger la información de una manera coherente y eficaz es necesario implementar un Sistema de Gestión

de Seguridad de la Información (SGSI). Este sistema es una parte del sistema global de gestión, basado en un análisis de los riesgos del negocio, que permite asegurar la información frente a la pérdida de:

- Confidencialidad: sólo accederá a la información quien se encuentre autorizado.
- Integridad: la información será exacta y completa.
- Disponibilidad: los usuarios autorizados tendrán acceso a la información cuando lo requieran. (Fernández, 2012)

A partir de esta definición se puede concluir que la seguridad de la información es un estado en el cual la información se encuentra libre de cualquier riesgo que haga peligrar la preservación de su confidencialidad, integridad, disponibilidad lo que se logra aplicando el conjunto adecuado de políticas, prácticas y normas capaces de mantener los pilares fundamentales antes expuestos.

Control de acceso en Sistemas de Información

El control de acceso representa un proceso indispensable en organizaciones cuyo funcionamiento depende de información digital con distintos niveles de sensibilidad. David Kim y Michael Solomon en el año 2010 definen el control de acceso como la verificación de si una entidad (una persona, vehículo, ordenador, etc..) solicitando acceso a un recurso tiene los derechos necesarios para hacerlo, ofrece la posibilidad de acceder a recursos físicos (por ejemplo, a un edificio, a un local, a un país) o lógicos (por ejemplo, a un sistema operativo o a una aplicación informática específica) (Kim, 2010).

El concepto de control de acceso en el que estará basado esta investigación será el definido por la Resolución No. 127 /2007 del Ministerio de la Informática y las Comunicaciones (MIC) de la República de Cuba. En él se expresa que “el control de acceso es un método que garantiza que solo tengan acceso a un sistema o a la información que éste contiene, aquellos debidamente autorizados para ello. Los mecanismos de control de acceso se implementan utilizando técnicas de software y de hardware y por lo general incluyen: identificación y autenticación de usuarios; limitación de acceso a ficheros, monitorización de las acciones de los usuarios y un sistema de auditoría” (2007). A pesar de la diversidad de los conceptos, en todas las literaturas como en esta, se incluyen de forma implícita los tres principales procesos del control de acceso:

- **Identificación**: es la acción por parte de un usuario de presentar su identidad a un sistema, generalmente se usa un identificador de usuario. Establece que el usuario es responsable de las acciones que lleve a cabo en el sistema.

- **Autenticación**: es la verificación de que el usuario que intenta identificarse es válido, usualmente se implementa con una contraseña en el momento de iniciar una sesión.
- **Autorización**: es un proceso para determinar si un sujeto identificado y autenticado tiene acceso al recurso solicitado. Dando la posibilidad de ejecutar operaciones específicas, dependiendo de sus derechos de acceso pre-configurados. La política de autorización debe ser gestionada por el administrador o agente de seguridad responsable de apoyar y llevar a cabo la política de seguridad en la organización.
- **Auditoría**: es el proceso de registro y análisis de todas las acciones ejecutadas por los sujetos sobre los recursos, a través de un SI. La auditoría es un aspecto crítico para identificar violaciones, debilidades, amenazas y predecir comportamientos y oportunidades de mejoras que apoyen la toma de decisiones en las organizaciones (Baryolo, 2012).

Modelos de control de acceso actuales

Modelos de control de acceso tradicionales

Existen dos tipos de modelos tradicionales de control de acceso, el modelo de Control de Acceso Discrecional o *Discretionary Access Control* (DAC), y el modelo de Control de Acceso Mandatario o *Mandatatory Access Control* (MAC).

- La característica principal del modelo DAC, es que el sujeto (sea un usuario o un proceso) del sistema, de manera autónoma puede otorgar su propio acceso hacia algún objeto (en su totalidad o parcialmente) a otros actores. Su implementación se realiza generalmente estableciendo una matriz de control de acceso al sistema. En esta matriz, las filas corresponden a los sujetos del sistema, las columnas corresponden a los objetos del sistema y las celdas representan a los derechos de acceso hacia los objetos por los sujetos (J., 2013).
- En el modelo de Control de Acceso Mandatario, una autoridad central está al mando de dar las decisiones de acceso a un sujeto que solicite el acceso hacia algún objeto o alguna información de los objetos. Con el fin de garantizar el acceso a los objetos y a la información que fluye entre ellos, el modelo de Control de Acceso Mandatario asigna una etiqueta de acceso a cada sujeto y objeto. Una etiqueta de acceso es un nivel de seguridad que se utiliza para asegurar el flujo de información entre los objetos y sujetos con una relación de dominación. Estas etiquetas de seguridad que se

utilizan para clasificar los objetos en función a la sensibilidad de la información que tienen. Las autorizaciones de los sujetos son los niveles de seguridad que se utilizan para reflejar la confiabilidad o las reglas de los sujetos (An access control model for cloud computing, 2014).

Modelo de Listas de Control de Acceso (ACL)

Las Listas de Control de Acceso (*Access Control List, ACL*), consisten en una tabla que le dice al sistema operativo qué permisos de acceso tiene un usuario sobre un objeto particular del sistema, por ejemplo, un fichero o un directorio. Cada objeto tiene un atributo de seguridad que identifica su ACL. Estas pueden ser difícil de gestionar en un entorno empresarial donde muchas personas necesitan tener diferentes niveles de acceso a distintos recursos (2014).

Modelo de Control de Acceso Basado en Atributos (ABAC)

El modelo de Control de Acceso Basado en Atributos, se basa en un conjunto de atributos asociados a un solicitante o a un recurso a ser visitado, con el fin de tomar las decisiones de acceso. Hay muchas maneras de definir o utilizar los atributos en este modelo. Un atributo puede ser un trabajo, fecha de inicio de un usuario, una ubicación de un usuario, un rol de un usuario o de todos ellos. Los atributos pueden también estar o no relacionados entre sí. Después de definir los atributos que se utilizarán en el sistema, cada atributo es considerado como un valor discreto, y los valores de todos los atributos se comparan con un conjunto de valores para un punto de decisión de una política de conceder o denegar algún acceso. En este tipo de modelo, un sujeto no tiene que ser conocido con anticipación por el sistema, sólo tiene que autenticarse en el sistema y luego proveer sus atributos. Por último, ellos describen que tener una política de seguridad que pueda funcionar con precisión con este tipo de modelo de control de acceso es vital, debido a que la política de seguridad es responsable de seleccionar a los atributos importantes que se utilizarán para tomar las decisiones de acceso (An access control model for cloud computing, 2014)

Modelo de Control de Acceso Basado en Roles (RBAC)

RBAC emplea el uso de roles predefinidos que llevan un conjunto específico de privilegios asociados a ellos y que se asignan. Por ejemplo, un sujeto al que le asignan al rol de administrador, tendrá acceso a un conjunto diferente de objetos que alguien a quien se le haya asignado el rol de analista. En este modelo, el acceso está predeterminado implícitamente por la persona que asigna los roles a cada individuo y explícitamente por el propietario del objeto al determinar el privilegio asociado con cada rol. En el punto de

una solicitud de acceso, el mecanismo de control de acceso evalúa la función asignada al sujeto que solicita acceso y el conjunto de operaciones que este rol está autorizado a realizar en el objeto antes de representar y aplicar una decisión de acceso.

Tenga en cuenta que un rol se puede ver como un atributo de sujeto que es evaluado por el mecanismo de control de acceso y alrededor de qué política de acceso a objetos se genera. A medida que la especificación RBAC ganó popularidad, hizo posible la administración central de las capacidades de control de acceso empresarial y redujo la necesidad de ACL (D.Ferraiolo, 2013).

Luego de realizarse un estudio y análisis de los diferentes modelos de control de acceso, se ha decidido utilizar el modelo de Control de Acceso Basado en Roles (RBAC), ya que permitirá una correcta administración de autorizaciones, el establecimiento de jerarquía de roles y la separación de responsabilidades, además permitirá que el control y mantenimiento de las políticas de acceso se manejen de una manera centralizada, lo que garantiza flexibilidad, separación de tareas, seguridad en el acceso a los recursos y a la información.

Tendencias actuales en los sistemas de control de acceso

Con la creciente migración de las empresas al control de acceso electrónico con el objetivo de mejorar la seguridad de su información y recursos, se ha visto el surgimiento de tendencias relacionadas a el desarrollo de este tipo de implementación, vinculadas principalmente a las demandas de mercado, a continuación, se mencionan algunas de esas tendencias.

Según una investigación realizada por la firma *Information Handling Services* (IHS) compañía pionera en el análisis de información global con sede en Londres, Reino Unido, se prevé que los ingresos derivados de la venta de cerraduras eléctricas y cerraduras electromagnéticas superarán los de las cerraduras mecánicas durante el 2017. En un informe sobre análisis del mercado, realizado en enero de 2014, IHS aseguró que, según sus proyecciones, los ingresos mundiales de estos productos aumentarán en tasas de crecimiento compuestos de 6,9% y 7,8%, respectivamente. En contraste, se proyecta un crecimiento menor para las cerraduras mecánicas de una tasa compuesta de 4,5% en el mismo período de tiempo. Adi Pavlovic, analista de control de acceso, protección contra incendios y seguridad de HIS, atribuye el aumento de la demanda de estos productos al incremento de la popularidad de los sistemas de acceso electrónicos. La firma asegura que las cerraduras electromagnéticas y las cerraduras eléctricas son los dispositivos de bloqueo eléctricos más comunes utilizados con los sistemas de control de acceso (Espinosa, 2017).

Edificios inteligentes y seguridad inteligente

El control de acceso también va más allá de la seguridad para proporcionar a los usuarios inteligencia y valor comercial. Esto se manifiesta a través del control de acceso en edificios inteligentes.

"Al ir más allá del ámbito de la seguridad física, el sistema de control de acceso necesita pagarse a sí mismo y agregar valor real mensurable a las empresas. Esto se puede lograr reduciendo los costos operativos o haciendo más mejorando las eficiencias operativas ", dijo Verner. "Un ejemplo reciente del despliegue de un edificio inteligente de CEM Systems es el nuevo e icónico proyecto *'One Albert Quay'* en Cork, Irlanda. Reconocido como el edificio más inteligente de Irlanda, *One Albert Quay* tiene todos sus sistemas críticos, incluidos iluminación, calefacción, energía, control de acceso, video, detección de incendios y extinción de incendios, todos conectados para permitir el uso inteligente de los datos (asmag, 2017).

Sin embargo, la seguridad sigue siendo una parte importante en un edificio inteligente y no debe ignorarse. "Desde un punto de vista de seguridad, el sistema de control de acceso y subsistemas integrados como video, fuego e intrusión. necesitan trabajar más inteligentemente para el usuario, con sistemas que comparten información y datos de manera precisa para optimizar el rendimiento total del edificio ", dijo Verner. "A medida que la industria se mueve más hacia medidas de seguridad preventivas, el papel de la integración de sistemas y la recopilación unificada de análisis a partir de la construcción de 'datos' se vuelve aún más profundo. Después de todo, la seguridad inteligente no se trata de capturar el evento después de que haya sucedido. 2017 verá el uso creciente de datos y análisis de construcción colaborativa para analizar las vulnerabilidades antes de que sucedan " (asmag, 2017).

Plataformas unificadas

Uno de los mayores cambios ha sido el paso de integraciones a unificaciones, a plataformas unificadas. La unificación proporciona acceso e información de video que reduce el costo total de propiedad con una sola solución de servidor junto con capacidades de funciones mejoradas. Un ejemplo de estas capacidades sería la capacidad de tener conjuntos más profundos de capacidades de análisis e informes para los clientes fuera de los conjuntos de datos combinados (Ouellete, 2014).

Migración a control de acceso *Internet Protocol*

Los sistemas electrónicos de bloqueo electrónico se vuelven aún más atractivos con la transición al control de acceso *Internet Protocol* (IP), entre cuyos beneficios se cuentan la operación simplificada del sistema, la

expansión y adaptación a medida y la posibilidad de integrar un Sistema de Control de Acceso Físico (PACS, por sus siglas en inglés) con muchas otras soluciones que comparten la misma red.

La mayoría de empresas e instituciones actuales tiene instalada una amplia variedad de sistemas generalmente dispares y aislados, desde sistemas de seguridad, control de acceso y video vigilancia, hasta sistemas de respuesta a incidentes, detección perimetral y monitoreo de alarmas. Aunque normalmente estos sistemas no pueden compartir fácilmente la información (si es que pueden hacerlo en absoluto), existen sinergias naturales entre cada uno de ellos. Las soluciones IP facilitan su integración y brindan la oportunidad de tener un solo sistema que puede ser mucho más que la suma de sus partes separadas (Espinosa, 2017).

Hardware y tecnologías utilizados en los sistemas de control de acceso actuales

A continuación, se muestra un resumen de tecnologías y hardware más utilizados en la implementación de sistemas de control de acceso actuales:

Tecnología de tarjetas

Tarjetas de proximidad, llamada también identificación por radiofrecuencia (RFID) es un método de identificación automática sin contacto; es la tecnología más nueva y de más rápido crecimiento en el segmento de identificación automática en la industria. RFID permite identificación automática, localización y monitoreo de personas, objetos y animales en una infinidad de aplicaciones (Baechli, 2005).

Estas tarjetas no tienen desgaste, ya que al ser una tecnología de identificación por radiofrecuencia (RFID), solamente hay que acercarla al lector y por lo tanto no existe el desgaste por rozamiento.

La tarjeta puede ser leída aún si la misma no es removida de la cartera o billetera y a través de la mayoría de otros materiales no metálicos. La orientación de la tarjeta y del lector no es crítica y el contacto con monedas o llaves no alterará su código, ni impedirá una lectura precisa y exacta.

Las tarjetas de proximidad no tienen partes móviles, ni contactos eléctricos que limpiar, ni uso o desgaste mecánico, tampoco cabezas lectoras que mantener y es resistente a los actos de vandalismo, ya que, desde el punto de vista de la seguridad, lo más importante es que no puede ser duplicada. Esto otorga a los sistemas de control de accesos implementados con esta tecnología un grado máximo de seguridad. Hoy en día, las tarjetas de proximidad, es una de las tecnologías más moderna y efectiva, por su practicidad y bajo

costo de mantenimiento. Tiene un costo medio, sin embargo, su duración hace que resulte, la más económica, porque no requiere recambios por desgastes (Baechli, 2005).

Existen dos tipos de tarjetas de proximidad: las tarjetas pasivas y las activas. Las tarjetas pasivas toman la energía generada por el lector para emitir su código, son más livianas, más económicas y más durables y tienen un alcance hasta 70 cm. Las tarjetas activas tienen incorporada una batería de duración limitada y no recambiable, su ventaja radica en su rango de lectura el cual puede llegar a 1,5 m contra los 70 cm. alcanzables hoy por las tarjetas pasivas (Baechli, 2005). La ilustración número 1 muestra una tarjeta de proximidad (Ver Anexo 1).

Para la lectura de las tarjetas de proximidad, el lector de proximidad constantemente transmite una señal de radiofrecuencia fija de bajo nivel, la cual provee energía a la tarjeta de proximidad. Cuando la tarjeta es presentada a cierta distancia del lector, la señal de RF es absorbida por una pequeña bobina dentro de la tarjeta y energiza al chip de la tarjeta, el cual contiene un único código de identificación. Una vez energizada, la tarjeta transmite el código al lector. El proceso es completado en milisegundos (Baechli, 2005). La ilustración número 2 (Ver Anexo 1) muestra un lector de tarjetas de proximidad

Tarjetas de código de barras

Las tarjetas con código de barras son un método simple, fácil y económico para codificar información numérica y alfanumérica, legible por lectores electrónicos. Un código de barras es la representación gráfica**,** mediante barras y espacios, de un conjunto de caracteres.

El código de barras es uno de los sistemas de identificación electrónica más comunes y se implementa sobre todo en el comercio para la identificación de productos, clientes y empleados**.** Muchos comercios optan por tarjetas de PVC (Poli cloruro de vinilo) con código de barras para la fidelización de clientes porque al disponer ya de lectores de código de barras para la gestión de su negocio, es una opción muy económica (2017). Las ilustraciones número 3 (Ver Anexo 1) muestra una tarjeta y lector de código de barras respectivamente.

Sistemas de llaves electrónicas

La llave electrónica, es una pastilla electrónica encapsulada en acero inoxidable de unos 16 mm de diámetro, que se transportan con un soporte plástico de unos 5 cm de largo con un ojalillo en su parte superior para poder colgarlo en un llavero. Brindan un muy alto nivel de seguridad, ya que son altamente

resistentes al desgaste, siendo ideales para ambientes industriales en donde la probabilidad de falla, vandalismo o sabotaje sea alta, aunque no son recomendables para ambientes con alto grado de generación de corriente estática, por ejemplo, oficinas con mucha alfombra y ambientes muy secos.

Las llaves son duraderas y funcionan sin pila ni batería. Cada llave tiene un código único (más de 280 mil millones de combinaciones), con esta tecnología, evita la posibilidad de duplicarlas, haciéndolas muy confiables. En precio son unos de los medios más caros, aunque en relación nunca se desgastan, como puede suceder con una tarjeta, con lo cual a largo plazo resulta conveniente (Baechli, 2005). Las ilustraciones número 5 y 6 (Ver Anexo 1) muestra una llave y lector electrónico respectivamente.

Sistemas biométricos

Las tecnologías biométricas fueron utilizadas en sus orígenes con fines legales, básicamente de investigación criminal. Pero el avance de las TIC en todos los órdenes ha ampliado su utilización con otros fines. Los sistemas biométricos, en general, están compuestos de dispositivos para recopilar datos en formato digital; algoritmos de procesamiento de los datos recolectados, que efectúan control de calidad de los datos y van conformando las plantillas biométricas. Las plantillas se almacenan en una base de datos contra la cual se comparan los datos ingresantes en la posterior actividad de verificación. El proceso de comparación permite realizar el cotejo de los nuevos datos con los datos almacenados. Finalmente, un proceso de decisión recomienda tomar una decisión en el proceso de reconocimiento, a partir de los resultados del componente de coincidencia (Etchart, 2011).

Tipos de sistemas biométricos:

-Huellas dactilares

-Reconocimiento facial

-Reconocimiento de geometría palmar

-Reconocimiento de voz

-Reconocimiento de escritura

-Reconocimiento del iris

-Reconocimiento de retina

-Reconocimiento del ácido desoxirribonucleico(ADN)

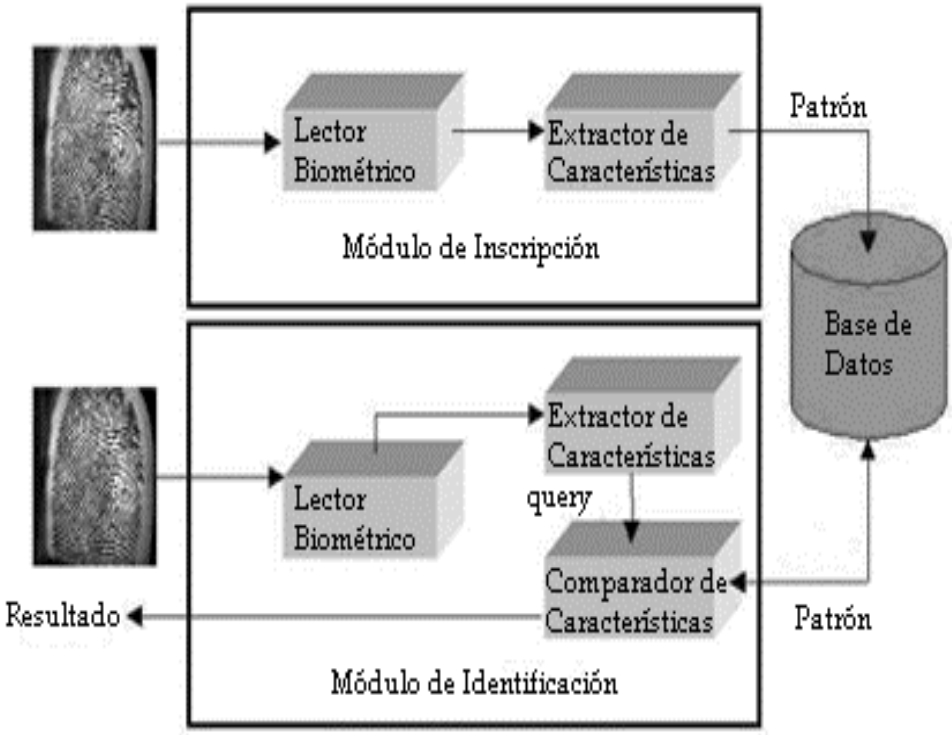


Ilustración 1: Arquitectura de un sistema biométrico.

Valoración del hardware y las tecnologías estudiadas

Luego de realizar un análisis del hardware y las tecnologías estudiados se concluye que, debido a los altos costos de adquisición de los mismos, así como la necesidad de realizar cambios estructurales en las áreas donde se desee implantar, teniendo en cuenta la situación económica del país, se decide para la presente investigación desechar el uso de hardware y tecnologías avanzados para el control de acceso. A pesar de lo dicho se tiene en cuenta que el uso de tarjetas de proximidad podría formar parte de una solución en el futuro, valorando las ventajas que presentan mencionadas anteriormente.

Sistemas de control de acceso a nivel internacional

WIN-PAK

El software WIN-PAK de Honeywell proporciona una gama de soluciones que abarcan desde simples soluciones de control de accesos a soluciones de seguridad totalmente integradas. La interfaz de WIN-PAK

permite a los integradores instalar una única solución de software para satisfacer todas las necesidades de seguridad de sus clientes, es adecuado para los sistemas con una estación de trabajo única que requieren funcionalidad de accesos sin ninguna integración. Este software de control de accesos gestiona de manera efectiva los controladores y permite la expansión y ampliación para Integración (Honeywell, 2018).

Prowatch

Diseñada para enfrentar los desafíos de los entornos exigentes de hoy en día, la suite de administración de seguridad Pro-Watch® de Honeywell brinda la flexibilidad, escalabilidad y control requeridos para una administración de seguridad integral. Pro-Watch equipa a las organizaciones con las herramientas adecuadas para proteger a las personas, proteger los activos y garantizar el cumplimiento normativo mediante la combinación de control de acceso, video digital, intrusión y otras funciones en un sistema poderoso.

La integración con el control de acceso y los sistemas de video de Honeywell y de terceros fabricantes hace posible aprovechar el hardware instalado existente a medida que el sistema crece. Las opciones de hardware y software modulares hacen que sea fácil y rentable ampliar un sistema para mantenerse al día con las crecientes necesidades comerciales (Honeywellintegrated, 2018).

Brivo

Brivo Inc., un proveedor de sistemas de control de acceso físico basado en la nube, lanzó Brivo Mobile Pass, una solución de credenciales digitales que permite a los usuarios desbloquear puertas con sus teléfonos inteligentes. Como una mejora del sistema de control de acceso Brivo OnAir existente, los clientes pueden distribuir inmediatamente las credenciales móviles a toda su población de usuarios sin ningún cambio de equipo en el local.

Disponible ahora en iOS y Android, Mobile Pass revoluciona la seguridad física al entregar inmediatamente control de acceso a través de teléfonos inteligentes sin tener que instalar nuevos lectores de puertas. Brivo Mobile Pass sirve como un complemento móvil para tarjetas de claves físicas y lectores, y es completamente interoperable con las tecnologías existentes de lectores de puertas (Sdmmag, 2015).

Protection 1

Este ofrece una gama de sistemas de control de acceso electrónico tradicionales y alojados de los principales fabricantes. Las aplicaciones alojadas en la web con tecnología de Brivo permiten una

administración completa de su sistema en la web, de una a cientos de ubicaciones. Los sistemas Winpak y ProWatch de Honeywell cubren una gama de aplicaciones en un entorno de instalaciones independientes o en red. Los controladores especiales habilitados para IP pueden permitir que su sistema de control de acceso se extienda fácilmente a todas las partes de una empresa con el grado de seguridad apropiado en cada puerta. Protection 1 puede actualizar su sistema existente o crear un nuevo sistema escalable con compatibilidad incorporada para todas las tecnologías de tarjetas, incluidas las tarjetas inteligentes, dispositivos biométricos y de control de acceso inalámbrico. Sus soluciones de seguridad de control de acceso van desde simples sistemas autónomos de control de acceso a la seguridad empresarial hasta complejos sistemas alojados con cientos de lectores de tarjetas integrados con dispositivos de sistema de alarma antirrobo y tarjetas de video (Protection1, 2018).

Tyco

Los sistemas biométricos de control de acceso de Tyco Integrated Fire & Security permiten llevar un control de acceso personalizado y discreto. Sus escáneres y cerraduras de identificación eliminan la necesidad de tarjetas o códigos numéricos para el control de acceso, mediante el uso de una sofisticada tecnología biométrica para mejorar la seguridad de sus instalaciones y su gente. También, ofrecen servicios de evaluación de control de acceso. Mediante la instalación de servidores biométricos, lectores de huellas digitales y otras soluciones de identificación facial o del iris (Tyco, 2017).

Sistemas de control de acceso a nivel nacional

XYMA SMART ACCESS

Es una solución informática para la monitorización, registro y control del tránsito de personas a través de puntos de acceso en instalaciones, permitiendo su configuración con un alto nivel de seguridad, combinando el uso de tarjetas de identificación, información biométrica y códigos de acceso. puede ser de utilidad en: Centros e instituciones que requieran de altos niveles de seguridad en el acceso a sus dependencias. De igual forma, es útil y práctico en pequeñas locaciones, organizaciones, negocios, hoteles, comercios y complejos de apartamentos residenciales (Datys).

Xyma Smart Access permite (Datys):

- El trabajo con las tarjetas de proximidad de baja frecuencia (125 KHz) y las tarjetas inteligentes sin contacto (MIFARE, 13.56 MHz).

- Construir el control de acceso a la medida de los requerimientos del cliente.
- Entrenamiento para usuarios y emisores de tarjetas.
- Procedimientos para emitir tarjetas de identidad con seguridad y asegurar que los documentos de identidad sean emitidos solamente por la entidad autorizada para expedir dichas tarjetas, y que solamente sean emitidos documentos de identidad para las personas correctas.
- Controles de seguridad que provean acceso a la información contenida en el documento de identificación solamente a observadores debidamente autorizados para ello.
- Un proceso de autenticación que implemente la cadena de confianza previamente establecida, verificando la identidad de los portadores del identificador y la legitimidad de las tarjetas de identidad y sus credenciales.

Xyma Smart Access tiene como características (Datys):

- Es compatible con el equipamiento de diferentes fabricantes de tecnología para el control de acceso. (L-1 Identity Solutions, Infinias, ZKSoftware).
- Se adapta eficaz y adecuadamente a las necesidades de cualquier entorno de trabajo que requiera seguridad en el acceso a sus instalaciones y registro de asistencia de empleados.
- Contienen los siguientes módulos: Monitor, Registro de Asistencia, Control de Acceso.
- Un proceso de registro seguro, que establece la entidad de cada individuo y que determina que la persona está autorizada para utilizar los privilegios o servicios que están siendo brindados.

XABAL IDBIOACCESS

Sistema de control de acceso e identificación

Sistema de Control de Accesos que permite verificar accesos mediante lectura de código de barras, búsqueda manual de los datos, comparación de las huellas dactilares con las almacenadas en la base de datos y/o en el código QR (del inglés *Quick Response Code*, "código de respuesta rápida") de la credencial. Permite la integración con distintos dispositivos de captura de huellas dactilares e impresoras de documentos ().

Sistema de control de acceso e interbloqueo para el Centro de Inmunología Molecular

La solución consiste en un sistema compuesto por dos elementos fundamentales: el controlador de puerta y el controlador de interbloqueo. Ambos son tarjetas electrónicas con finalidades diferentes dentro del sistema (Sistema de control de acceso e interbloqueo para el Centro de Inmunología Molecular, 2013).

La función del controlador de puerta es, como su nombre lo indica, el control de las puertas. Un controlador de este tipo puede controlar un máximo de dos puertas. De esta forma se encarga de recibir la solicitud de acceso que viene del elemento de identificación, determinar si el código arribado tiene acceso por la puerta especificada según previa configuración y, de ser positiva o válida, entonces desbloquear la puerta en cuestión (Sistema de control de acceso e interbloqueo para el Centro de Inmunología Molecular, 2013).

Cabe destacar que éste es capaz de recibir la codificación enviada desde cualquier lector que implemente protocolo Wiegand formato 26. Este protocolo es abierto y se considera un estándar dentro de los sistemas de control de acceso. De esta forma se garantiza la comunicación con una amplia gama de dispositivos de lectura que existen en la actualidad (Sistema de control de acceso e interbloqueo para el Centro de Inmunología Molecular, 2013).

Biomesys

Sistema de Control Biométrico de Asistencia.

Biomesys Control de Asistencia es un sistema que aprovecha las bondades de las tecnologías que aplican la biometría, para registrar los eventos de asistencia en una organización por medio de la identificación de los empleados y de la autenticación de su identidad mediante un censor biométrico de huellas dactilares. A partir de la captura de identificaciones biométricas únicas, el sistema se convierte en un generador de datos altamente confiable por su bajo o casi nulo nivel de vulnerabilidad por la suplantación de identidad. De utilidad para los que pretenden apoyarse en un instrumento sencillo con un enfoque flexible y adaptable a la estructura funcional de la organización, que propone herramientas personalizables para el control y la toma de decisiones en el área de los recursos humanos ().

COSMO

COSMO, Sistema de Control de Acceso a los Laboratorios o Agrupaciones de Computadoras(AC), que es capaz de controlar los aspectos relacionados con la interacción de los usuarios con los laboratorios o AC, tales como la entrada y salida de los usuarios, las horas de inicio y fin de sesión de las computadoras, lista

de procesos activos, entre otros. COSMO permite controlar de forma integral el uso de laboratorios o AC, generando además estadísticas que sirven de apoyo a la toma de decisiones (Beoto, 2010).

COSMO está basado en el sistema Cronos: Gestor de Entornos Informativos Personalizados, se define una estructura Cliente-Servidor, implementada de forma modular, en el Servidor se encuentran módulos servidores que interactúan con diferentes orígenes de información (base de datos, servidor de aplicaciones, páginas web, etc.), que a su vez se comunican con los módulos homólogos en el cliente (Beoto, 2010).

Sistema de control de acceso a comedores

El sistema de control de acceso a comedores, desarrollado en el Centro de Informatización de Entidades en la facultad 3, administra la asignación y distribución de los comensales de la UCI, en cada una de las puertas de los complejos comedores. A través de este sistema se facilita el proceso de acceso por parte de los comensales de la universidad a cada una de las puertas de los comedores disponibles. Es un sistema seguro, que basa sus permisos en usuarios definidos pertenecientes al dominio uci.cu y dado una dirección IP que posee la *Personal Computer* (PC) donde se va a trabajar. Las funcionalidades que posee son: Gestionar evento, Gestionar grupos, Gestionar distribución, Distribuir grupos, y Asignar accesos (Escalona, 2016).

Valoración de los sistemas estudiados:

A partir del estudio realizado a los sistemas de control de acceso estudiados se evidencian los siguientes resultados:

Con respecto a los sistemas estudiados a nivel nacional, podemos concluir que no constituyen soluciones que satisfagan las necesidades requeridas debido que están orientados a fines específicos y no cumplen con las características genéricas y de integralidad con los otros sistemas de la UCI, por otra parte, los sistemas estudiados a nivel internacional integran tecnologías muy avanzadas con el objetivo de solucionar funcionalidades complejas, estas tecnologías resultan poco óptimas teniendo en cuenta sus altos costos en conceptos de hardware y licencias.

No obstante, a partir del estudio realizado se lograron identificar varias características y funcionalidades de los sistemas de control de acceso:

- El diseño de un sistema que se ajuste a la necesidad del usuario.

- La capacidad de generar informes de eventos, útil a la hora de extraer históricos sobre las incidencias en los locales de trabajo.
- El desarrollo de una base de datos centralizada, que responda a todas las necesidades de los diferentes módulos del producto.
- La idoneidad del desarrollo de la aplicación en un entorno web, debido a un conjunto de ventajas como las posibilidades de cambiar el hardware y migrar de un sistema operativo a otro sin que se vea afectado el funcionamiento de las aplicaciones de servidor.

Metodologías de desarrollo

"Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo" (Gómez, 2010).

Metodologías tradicionales y Metodologías ágiles

Diversos autores coinciden en señalar algunos requisitos que deben tener las metodologías de desarrollo:

- Visión del producto.
- Vinculación con el cliente.
- Establecer un modelo de ciclo de vida.
- Gestión de los requisitos.
- Plan de desarrollo.
- Integración del proyecto.
- Medidas de progreso del proyecto.
- Métricas para evaluar la calidad.
- Maneras de medir el riesgo.
- Como gestionar los cambios.

- Establecer una línea de meta.

En tiempos recientes, han surgido las metodologías ágiles, como una alternativa, una reacción a las metodologías tradicionales y principalmente a su burocracia. Brooks, en su mítico libro The Mythical Man Month (El mes del hombre mítico), expone las primeras ideas que se plantean en las metodologías ágiles, gran parte de ellas, responden al sentido común (Gómez, 2010). Las características de ambas metodologías quedan resumidas en la siguiente tabla:

Metodologías Agiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Mas artefactos
Pocos roles	Mas roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Ilustración 2: Características de las metodologías ágiles y tradicionales.

A partir las características expuestas anteriormente se concluye que teniendo en cuenta el escenario de la presente investigación resulta objetivo el uso una metodología ágil, debido a que el desarrollo de la aplicación será realizado por una sola persona, tanto en las labores de análisis como diseño e implementación, considerando también que no existe un contrato prefijado y que está metodología es más flexible con respecto al cambio. A continuación, se realizará un estudio sobre las diferentes metodologías agiles con el objetivo de seleccionar la más adecuada para ser utilizada en la presente investigación.

Programación Extrema (EXTREME PROGRAMMING, XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en realimentación 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 cambiantes, y donde existe un alto riesgo técnico (Letelier, 2003).

Scrum

Su nombre no corresponde a una sigla, sino a un concepto deportivo, propio del rugby, relacionado con la formación requerida para la recuperación rápida del juego ante una infracción menor. Su primera referencia en el contexto de desarrollo data de 1986, cuando Takeuchi y Nonaka utilizan el Rugby para definir un nuevo enfoque en el desarrollo de productos, dirigido a incrementar su flexibilidad y rapidez, a partir de la integración de un equipo interdisciplinario y múltiples fases que se traslapan entre sí. La metodología Scrum para el desarrollo ágil de software es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento. Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas (Cadavid, 2013).

Crystal

La familia de metodologías Crystal se basa en los conceptos de Proceso Racional Unificado o Rational Unified Process(RUP) y está compuesta por Crystal Clear, Crystal Yellow, Crystal Orange y Crystal Red; el nivel de opacidad del color en el nombre indica un mayor número de personas implicadas en el desarrollo, un mayor tamaño del proyecto y, por lo tanto, la necesidad de mayor control en el proceso. La filosofía de Crystal define el desarrollo como un juego cooperativo de invención y comunicación cuya meta principal es entregar software útil, que funcione, y su objetivo secundario, preparar el próximo juego. Los valores compartidos por los miembros de la familia Crystal están centrados en las personas y en la comunicación.

Sus principios indican que: el equipo puede reducir trabajo intermedio en la medida que produce código con mayor frecuencia y utiliza mejores canales de comunicación entre las personas; los proyectos evolucionan distinto con el tiempo por lo que las convenciones que el equipo adopta tienen que adecuarse y evolucionar; los cambios en el cuello de botella del sistema determinan el uso de trabajo repetido; y el afinamiento se realiza sobre la marcha. Existen dos reglas que aplican para toda la familia Crystal. La primera indica que los ciclos donde se crean los incrementos no deben exceder cuatro meses; la segunda, que es necesario realizar un taller de reflexión después de cada entrega para afinar la metodología (Cadavid, 2013).

AUP

AUP (Agile Unified Process o Proceso Unificado Ágil, en español) es un enfoque de modelado híbrido creado por Scott Ambler cuando combinaba el Proceso Racional Unificado (RUP) a métodos ágiles (AM). Scott Ambler trabaja para el grupo IBM Methods como el líder de práctica para el desarrollo ágil. Al combinar RUP con AM, Ambler creó un marco de procesos sólido que se puede aplicar a todo tipo de proyectos de software, grandes o pequeños. Los métodos ágiles proporcionaron valores, principios y prácticas para AUP. El manifiesto ágil muestra cuáles son estos valores y principios. El manifiesto describe cuatro declaraciones de valores para desarrollo ágil. Estos valores incluyen individuos y sus acciones, entregando software en funcionamiento, colaboración con el cliente y respondiendo al cambio. Los principios descritos en el manifiesto incluyen satisfacer al cliente a través de entrega de software inicial y continua, preparándose para el cambio, desarrolladores y negocios colaborando a lo largo del proyecto, creando proyectos a través de individuos motivados, utilizando los medios más efectivos para transmitir información en conversaciones cara a cara, atención a la excelencia técnica, simplicidad, uso de equipos auto organizados, reflexión y ajustes regulares. Cuando Ambler creó la AUP, centró el diseño en torno a los siguientes principios:

- La mayoría de las personas no leerán la documentación detallada. Sin embargo, necesitarán orientación y capacitación de vez en cuando.
- El proyecto debe describirse simplemente en unas pocas páginas.
- La AUP se ajusta a los valores y principios descritos por Agile Alliance.
- El proyecto debe enfocarse en entregar valor esencial en lugar de innecesario características.
- Los desarrolladores deben tener la libertad de usar las herramientas más adecuadas para la tarea en cuestión, en lugar de cumplir con un edicto.

- AUP se adapta fácilmente a través de herramientas comunes de edición de HTML (Edeki, 2013).

AUP-vUCI

Esta metodología de desarrollo se denomina Proceso Unificado Ágil versión Universidad de las Ciencias Informáticas (AUP-vUCI). Esta presenta tres fases que se ejecutan en orden: inicio, ejecución y cierre; el ciclo de vida de los proyectos se organiza en siete disciplinas: modelado de negocio, requisitos y análisis y diseño, implementación; mientras que en el caso de prueba se desagrega en: pruebas internas, de liberación y aceptación; las otras tres disciplinas de AUP asociadas a la parte de gestión para la variación UCI, se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2: gestión de la configuración (CM), planeación de proyecto (PP) y monitoreo y control de proyecto (PMC). Esto se debe a la certificación del nivel 2 de CMMI-DEV que tiene la universidad en la actividad de desarrollo de producción. Para el modelado de negocio se propone tres variantes: casos de uso del negocio (CUN), descripción de procesos de negocio(DPN), o modelo conceptual (MC), y para encapsular los requisitos: casos de uso del sistema (CUS), historias de usuario(HU) y descripción de requisitos por procesos (DRP). De ahí surgen cuatro escenarios para modelar el sistema en los proyectos:

- Escenario No. 1: Proyectos que modelen el negocio con CUN. Solamente pueden modelar el sistema con CUS. CUN + MC = CUS.
- Escenario No. 2: Proyectos que modelen el negocio con MC solamente pueden modelar el sistema con CUS. MC = CUS.
- Escenario No. 3: Proyectos que modelen el negocio con DPN solamente pueden modelar el sistema con DRP. DPN + MC = DRP.
- Escenario No. 4: Proyectos que no modelan el negocio solamente modelan el sistema con HU (Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos., 2017).

Elección de la metodología

La metodología seleccionada para guiar el proceso de desarrollo del software será AUP-vUCI, debido a las ventajas que posee por ser una metodología de desarrollo ágil, que permite la simplicidad, al centrarse en actividades de alto valor esenciales para el desarrollo, la adaptación al uso de herramientas independientes y la capacidad de reunir en una única disciplina las etapas de Modelado de Negocio, Requisitos y Análisis de Diseño. Otro motivo por el cual se eligió esta metodología es por ser nativa de la UCI, por lo que su

implementación resultará más sencilla a los implicados en el desarrollo del software debido a que ya están relacionados con la misma.

Herramientas y tecnologías a utilizar

Lenguaje de modelado

El lenguaje unificado de modelado o UML (*Unified Modelillg Language*) es el sucesor de la oleada de métodos de análisis y diseño orientados a objetos (OOA&D) que surgió a finales de la década de 1980 y principios de la siguiente.

El UML unifica, sobre todo, los métodos de Booch, Rumbaugh (OMT) y Jacobson, pero su alcance llegará a ser mucho más amplio. En estos momentos el UML está en pleno proceso de estandarización con el OMG (Object Management' Group o Grupo de administración de objetos) (Fowler, 1999).

Decimos, pues, que el UML es un lenguaje de modelado, y no un método. La mayor parte de los métodos consisten, al menos en principio, en un lenguaje y en un proceso para modelar. El lenguaje de modelado es la notación (principalmente gráfica) de que se valen los métodos para expresar los diseños. El proceso es la orientación que nos dan sobre los pasos a seguir para hacer el diseño (Fowler, 1999).

Lenguaje de programación

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, http://www.python.org/, y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional. El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables (L.Drake, 2009). La versión que se utilizará es la 3.5.

Framework (Marco de trabajo) de desarrollo

Django es un *framework* web Python de alto nivel que permite el desarrollo rápido de sitios web seguros. Construido por desarrolladores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y pago (Mozilla, 2018).

Django fue desarrollado inicialmente entre 2003 y 2005 por un equipo web que era responsable de crear y mantener sitios web de periódicos. Después de crear varios sitios, el equipo comenzó a factorizar y reutilizar muchos códigos y patrones de diseño comunes. Este código común se convirtió en un marco de desarrollo web genérico, que fue de código abierto como el proyecto "Django" en julio de 2005. Django ha seguido creciendo y mejorando, desde su primer lanzamiento importante (1.0) en septiembre de 2008 hasta la versión 1.11 (2017) recientemente publicada. Cada lanzamiento ha agregado nuevas funcionalidades y correcciones de errores, que van desde soporte para nuevos tipos de bases de datos, plantillas de motores y almacenamiento en caché, hasta la adición de clases y funciones de vista "genéricas" (que reducen la cantidad de código que los desarrolladores tienen que escribir una serie de tareas de programación) (Mozilla, 2018). La versión de Django que se utilizará es la 1.10.3.

IDE (Integrated Develop Environment o Entorno de Desarrollo Integrado)

PyCharm presenta un editor de código inteligente que comprende los detalles de Python y proporciona productividad que incluye: formateo automático de códigos, finalización de código, refactorizaciones, importación automática, navegación por código con un clic y más. Además, con la adición de rutinas avanzadas de análisis de código como base, estas características hacen de PyCharm una herramienta útil tanto para los desarrolladores profesionales de Python como para aquellos que recién están comenzando con la tecnología ().

Y debido a que PyCharm se basa en la plataforma IntelliJ, hereda el soporte de ese producto para la edición de los lenguajes JavaScript, HTML (Lenguaje de Marcas de Hipertexto) y CSS (hoja de estilo en cascada), entre otras características de las que se beneficiarán los desarrolladores web. Con estas capacidades, se espera que PyCharm se convierta en un IDE de Python líder incluso antes de su próxima versión principal, dijo JetBrains en un comunicado de prensa en PyCharm (). Se utilizará el PyCharm 2017 2.2.

Sistema Gestor de Base de Datos

PostgreSQL es un avanzado sistema de bases de datos relacionales basado en *Open Source*. Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades. PostgreSQL está bajo licencia BSD. Un sistema de base de datos relacionales es un sistema que permite la manipulación de acuerdo con las reglas del´algebra relacional. Los datos se almacenan en tablas de columnas y renglones. Con el uso de llaves, esas tablas se pueden relacionar unas con otras (Denzer, 2002).

En la jerga de bases de datos, PosgreSQL usa el modelo cliente/servidor. Una sesión en PostgreSQL consiste en ejecución de los siguientes procesos. El servidor, que maneja archivos de bases de datos, acepta conexiones a las aplicaciones cliente, y realiza acciones en la base de datos. El programa servidor de bases de datos se conoce como *postmaster*. La aplicación cliente, que necesita realizar operaciones en la base de datos. Las aplicaciones cliente pueden ser de la más diversa naturaleza: pueden ser aplicaciones de texto en una consola, aplicaciones gráficas, un servidor web que accede a la base de datos para mostrar una página, o herramientas especializadas de mantenimiento de bases de datos. Como es habitual en las aplicaciones cliente/servidor, el cliente y el servidor pueden estar en diferentes máquinas. En este caso, estos se comunican sobre una conexión de red TCP/IP (Denzer, 2002). La versión a utilizar es la 9.4.

Herramienta de Ingeniería de software

Visual Paradigm es una herramienta de software diseñada para que los equipos de desarrollo de software modelen el sistema de información comercial y administren los procesos de desarrollo. Visual Paradigm admite lenguajes y estándares de modelado de la industria clave como Unified Modeling Language (UML), SysML, SoaML, BPMN, XMI, etc. Ofrece herramientas de software completas que las empresas necesitan para la captura de requisitos, análisis de procesos, diseño de sistemas, diseño de bases de datos, y otros (). La versión a utilizar es la 8.0.

Las ventajas que proporciona Visual Paradimg para UML son:

- Dibujo. Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.

- Corrección sintáctica. Controla que el modelado con UML sea correcto.
- Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- Integración con otras aplicaciones. Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.
- Trabajo multiusuario. Permite el trabajo en grupo, proporcionando herramientas de compartición de trabajo.
- Reutilización. Facilita la reutilización, ya que disponemos de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.
- Generación de código. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta ().

Conclusiones del capítulo

Luego del análisis de los sistemas existentes, tecnologías y tendencias actuales fue posible arribar a las siguientes conclusiones:

- El estudio de los conceptos fundamentales en cuanto a sistemas de control de acceso y las características esenciales de los mismos, sentó las bases para el desarrollo del módulo de control de acceso para el SIPP.
- La investigación sobre los sistemas homólogos existentes, nacionales e internacionales, arrojó que dichas soluciones no son factibles para su utilización por el SIPP, pero tienen características comunes que deben tenerse en cuenta en la elaboración de la propuesta de módulo de control de acceso.
- El análisis de las herramientas, lenguajes y tecnologías a utilizar para la implementación del módulo de control de acceso para el SIPP, permitió una mejor comprensión de las mismas.

Capítulo 2: Análisis y diseño del módulo de control de acceso y registro de visitantes

En este capítulo se establecen las principales características de la solución propuesta. Se define y representa el modelo del dominio en el que se ilustrarán los conceptos del sistema planteado y las relaciones entre ellos. Se desarrollan las tres primeras fases de la metodología seleccionada(AUP-vUCI), modelado de negocio, requisitos y análisis, y diseño, presentándose los modelos generados.

Modelo de dominio

Un modelo del dominio, también llamado modelo conceptual, es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Es importante resaltar que un modelo del dominio no representa componentes de software. No se trata de un conjunto de diagramas que describen clases de software, u objetos software con responsabilidades (Larman, 2004).

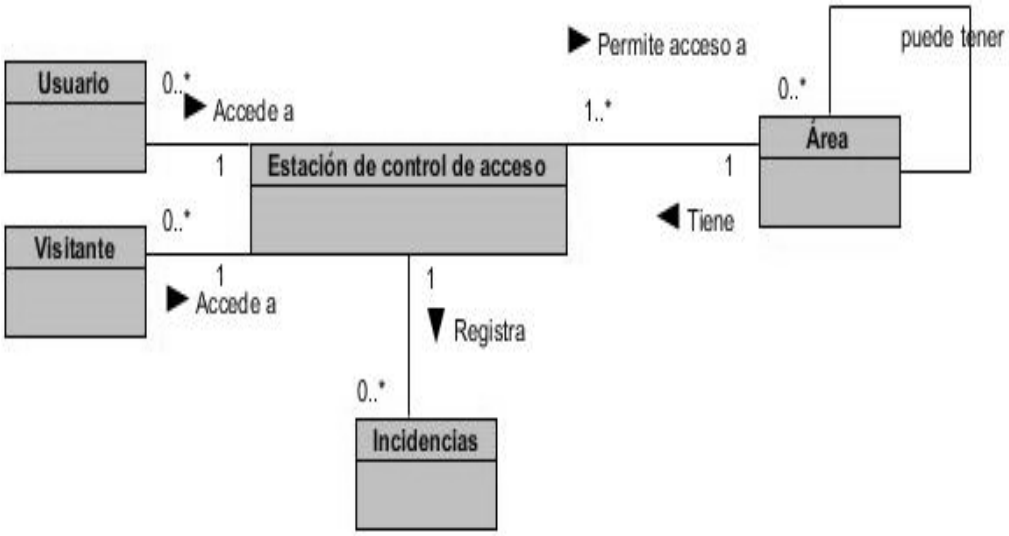


Ilustración 3: Modelo de dominio (Elaboración Propia).

Conceptos del Modelo de Dominio:

Usuario: Representa a todos los trabajadores o estudiantes de la institución.

Visitante: Representa a cualquier individuo ajeno a la institución.

Incidencias: Eventos que se produce en el transcurso de un asunto, un relato, etc., y que repercute en él alterándolo o interrumpiéndolo.

Estación de control de acceso: Representan los dispositivos y personas que realizarán el control de acceso.

Área: Distintos lugares de la institución.

Propuesta de solución

La solución propuesta tiene como objetivo el control de los datos de las personas, las reglas de acceso y las configuraciones de los puntos de acceso, será capaz de gestionar los visitantes y las distintas entradas realizadas a la institución, el sistema generará reportes sobre las incidencias ocurridas durante el proceso de control de acceso. Para dar cumplimiento al objetivo planteado se propone un módulo que se integrará al SIPP, el mismo estará basado en tecnologías web. Para poder acceder a las funcionalidades de este módulo el usuario deberá autenticarse en el sistema y solo si posee los permisos adecuados (previamente definidos) podrá acceder a las funcionalidades del módulo. Las funcionalidades básicamente estarán divididas en dos tipos:

Administración

Los usuarios con permisos de administración podrán crear nuevos puntos de acceso y reglas, definiendo sus características y comportamiento, además tendrán acceso a los reportes de eventos y las infracciones, las que serán registrada de forma automática cuando ocurran violaciones en el proceso de **“Control”.**

Control

Los usuarios con permisos de control podrán registrar el acceso tanto de trabajadores de la entidad como de los visitantes, y gestionar funcionalidades como la asignación de un solapín de visitantes.

Requisitos funcionales

RF1. Gestionar Puntos

RF2. Gestionar Reglas

RF3. Controlar Acceso

RF4. Registrar Visitantes

RF5. Listar Infracciones

RF6. Listar Accesos

RF7. Realizar reportes de Infracciones

RF8. Realizar reportes de Accesos

Requisitos no funcionales

Requisitos no funcionales de software para estaciones clientes

RnF1. Sistema Operativo Linux o Windows.

RnF2. Navegador web Mozilla Firefox v17.0 o superior, Google Chrome v20.0 o superior, o versiones actuales de Opera, Internet Explorer y Safari.

Requisitos no funcionales de software para estaciones servidores

RnF3. Sistema operativo Linux o Windows.

RnF4. PostgreSQL 9.4.

RnF5. Python 3.5.

Requisitos no funcionales de hardware para estaciones clientes

RnF6. PC Pentium 4 a 1GHz o superior, mínimo 512 MB de RAM.

Requisitos no funcionales de hardware para estaciones servidores

RnF7. PC Pentium 4 a 2 GHz o superior, mínimo 2 GB de RAM, 20 GB o superior de disco duro.

Requisitos no funcionales de usabilidad

RnF8. Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Además, debe ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencias una rápida adaptación.

RnF9. El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

- Conocimientos básicos relativos al uso de una computadora.
- Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

RnF10. El sistema será distribuido en idioma español.

RnF11. Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.

RnF12. El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación:

- Menús laterales y desplegables que permitan el acceso rápido a la información por parte de los usuarios. Restricciones en el diseño y la implementación.

RnF13. Lenguaje de Programación: Python.

RnF14. Plataforma de desarrollo: JetBrains PyCharm 2017 2.2.

RnF15.Framework: Django 1.10.

RnF16. Para el acceso a datos se utilizará Django ORM.

RnF17. Para el modelado de UML 2.0 se utilizará Visual Paradigm 8.0.

RnF18. Como Gestor de base de datos se utilizará PostgreSQL 9.4.

Requisitos no funcionalidades de seguridad

RnF19. La seguridad se define a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.

RnF20. Autenticación segura para acceder a la aplicación.

Requisitos no funcionalidades de fiabilidad

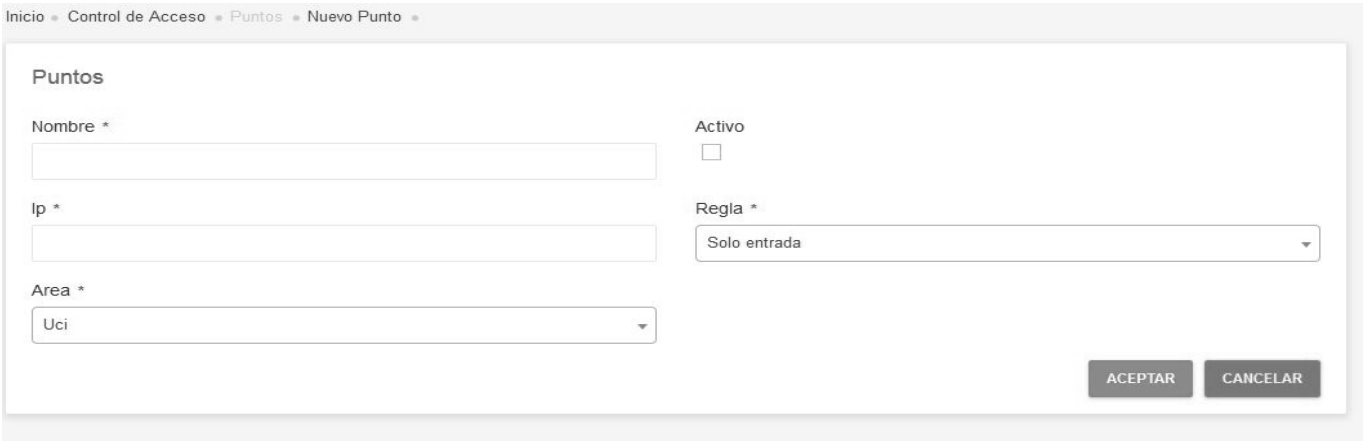
RnF21. Si ocurren errores en el sistema no se mostrarán detalles de los mismos al cliente

Historias de Usuario (HS)

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 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 implementarla en unas semanas (Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP), 2006). A

continuación, se muestra la Historia de Usuario (HU) número 1: Gestionar Punto, para más información ver Anexo 2.

Tabla 1: HU1 Gestionar Punto (Elaboración propia)

Número: HU-1	**Nombre de requisito:** Gestionar Punto
Programador: Christian M.Paneque	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo:** 32 horas
Riesgo: Alto	**Tiempo real:** 24 horas
Descripción: El usuario con los permisos adecuados será capaz de crear un nuevo punto, especificando su nombre, si estará activo o no, el IP desde donde se accederá al punto , las reglas del punto y el área a la que controla el acceso, tendrá la posibilidad de ver todos los puntos activos y eliminar o modificar un punto ya creado.	
Observaciones: Según los permisos del usuario tendrá diferente capacidad de edición.	
Prototipo de interfaz 	

Arquitectura del Sistema

El término ‘arquitectura’ es heredado de otras disciplinas de la ciencia. Se entiende por arquitectura a un conjunto de piezas de distintos tipos, que encajan entre sí y cumplen una función determinada. La arquitectura presenta además el impacto del cambio de una de las piezas (Vignaga, 2003).

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que

conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de marcos de trabajo basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores (Patrón Modelo-Vista-Controlador., 2012). En la siguiente ilustración se encuentra resumido su funcionamiento:

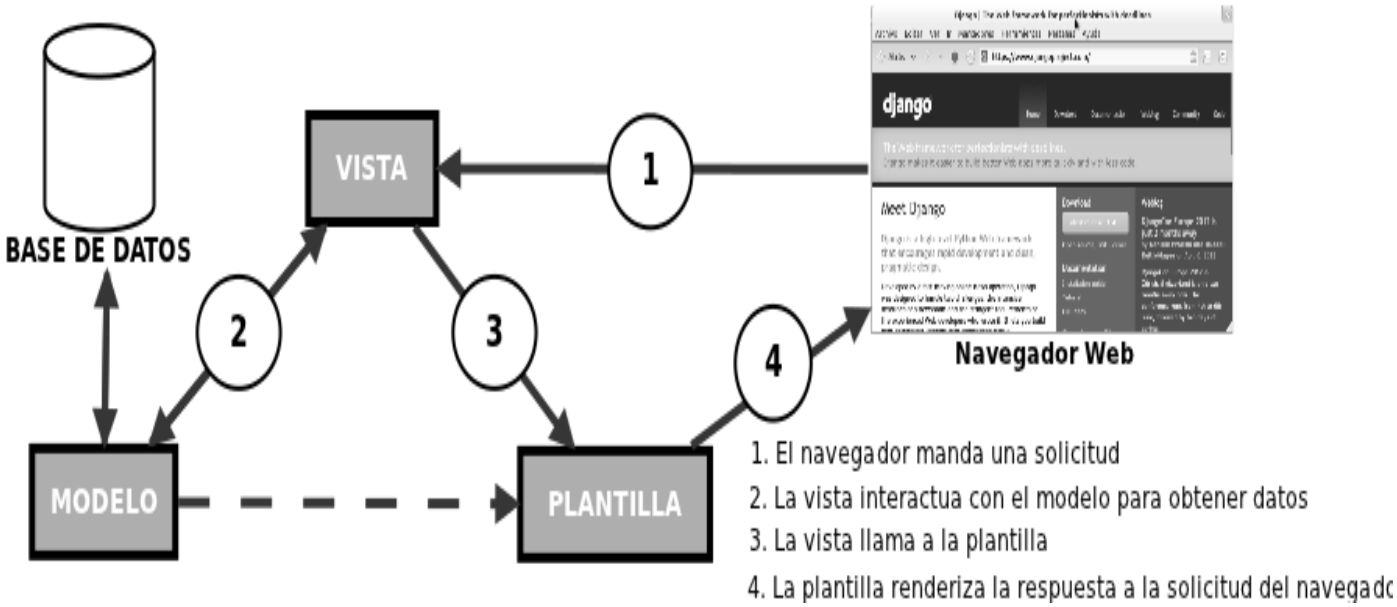


Ilustración 4: Arquitectura MVC.

El software desarrollado seguirá el patrón Modelo-Vista-Controlador debido a las capacidades mencionadas anteriormente y por ser la arquitectura utilizada por el marco de trabajo Django.

Patrones de diseño

Los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí (Larman, 2004).

En el diseño del módulo de control de acceso para el SIPP se tuvieron en cuenta los siguientes patrones

GRASP (Patrones Generales de Software para Asignación de Responsabilidades):

El patrón **Experto** posibilita una adecuada asignación de responsabilidades facilitando la comprensión del sistema, su mantenimiento y adaptación a los cambios con reutilización de componentes (Patrones Grasp y Anti-Patrones: un enfoque Orientado a Objetos desde Lógica de Programación, 2010). Este patrón se puede apreciar en el siguiente fragmento de código:

```
class tblPunto(models.Model):
    activo = models.BooleanField()
    nombre = models.TextField()
    fecha = models.DateTimeField(default=timezone.now, editable=False,)
```

Ilustración 5: Fragmento de código. Clase tblPunto (Elaboración propia)

La clase tblPunto (Ilustración 4) es la responsable de conocer la información para la creación de un objeto Punto, solo a través de ella se puede acceder a los atributos característicos de un “Punto”.

Patrón **Controlador**, este patrón tiene como objetivo asignar la responsabilidad a una clase de recibir o manejar un mensaje de evento del sistema generado por un actor externo, por lo general a través de una interfaz gráfica de usuario a la que accede un usuario para realizar ciertas operaciones en el sistema (Larman, 2004).

Patrón **Creador,** la instanciación de una clase es una de las actividades fundamentales en un sistema orientado a objetos. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, con lo que se logra menos dependencia y mayores oportunidades de reutilización de código (Larman, 2004).

```python
class VisitanteCreateView(RequiredSecurityMixin, SuccessMessageMixin, CreateView):
    need_login = True
    permission = RequiredSecurityMixin.CREATE
    model = tblVisitante
    template_name = 'crear.html'
    form_class = VisitanteForm
    success_url = reverse_lazy('insertar_visitante')
    success_message = "Visitante insertado: %(Nombre)s satisfactoriamente."

    def get_context_data(self, **kwargs):
        context = super(VisitanteCreateView, self).get_context_data(**kwargs)
        context['cancelar'] = self.success_url

        context['info_breadcrum'] = "Nuevo Visitante"
        context['info_panel'] = "Visitante"
        context['module'] = "Visitante"

        return context

    def form_invalid(self, form):
        messages.Error(self.request, 'Por favor corrija los errores.')
        return super(VisitanteCreateView, self).form_invalid(form)

    def get_success_url(self):
        register_logs(self.request, self.model, self.object.pk, force_str(self.object), ADDITION)
        return super(VisitanteCreateView, self).get_success_url()
```

Ilustración 6: Fragmento de código. Clase VisitanteCreateView (Elaboración propia)

El uso tanto de los patrones **Controlador y Creador** se evidencian en la clase VisitanteCreateView (Ilustración 5), esta clase es la encargada de interactuar como mediadora entra la interfaz y el modelo para ejecutar la inserción de los datos de un nuevo visitante, así como la creación de una instancia de la clase tblVisitante.

El patrón **Bajo Acoplamiento** es una medida de la fuerza con que una clase se relaciona con otras, porque las conoce y recurre a ellas; una clase con bajo acoplamiento no depende de muchas otras, mientras que otra con alto acoplamiento presenta varios inconvenientes: es difícil entender cuando está aislada, es ardua de reutilizar porque requiere la presencia de otras clases con las que esté conectada y es cambiante a nivel local cuando se modifican las clases afines (Patrones Grasp y Anti-Patrones: un enfoque Orientado a Objetos desde Lógica de Programación, 2010).

El uso de este patrón es inherente a la utilización del mismo marco de trabajo Django ya que este está diseñado para que sus distintas capas no deban “conocerse entre sí a no ser absolutamente necesario, Por ejemplo, el sistema de Templates (plantillas) no maneja información sobre las solicitudes Web y la capa de la base de datos no influye en la manera en que serán visualizados los datos.

Alta cohesión: en el diseño orientado a objetos, la cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento (clase o subsistema). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que colaboran entre sí y con otros objetos para simplificar su trabajo (Larman, 2004).

```
class ReglasCreateView(RequiredSecurityMixin, SuccessMessageMixin, CreateView):
    need_login = True
    permission = RequiredSecurityMixin.CREATE
    model = tblReglas
    template_name = 'crear.html'
    form_class = ReglasForm
    success_url = reverse_lazy('insertar_regla')
    success_message = "Regla insertada: %(nombre)s satisfactoriamente."
```

Ilustración 7: Fragmento de código. Clase ReglaCreateView (Elaboración propia)

```
class ReglasForm(forms.ModelForm):
    class Meta:
        model = tblReglas
        fields =['nombre','descripcion','chequea_entrada','activa','chequea_salida']
        widgets ={
            'nombre' : TextInput(attrs={'class': 'form-control tooltips'}),
            'descripcion': TextInput(attrs={'class': 'form-control tooltips'}),
            'chequea_entrada': CheckboxInput(attrs={'class': 'form-control tooltips'}),
            'activa': CheckboxInput(attrs={'class': 'form-control tooltips'}),
            'chequea_salida': CheckboxInput(attrs={'class': 'form-control tooltips'}),
```

Ilustración 8: Fragmento de código. Clase ReglasForm (Elaboración propia)

En la solución desarrollada es evidente la alta cohesión de las clases, como podemos apreciar entre las clases ReglasCreateView (Ilustración 6) y ReglasForm (Ilustración 7). En este caso la clase ReglasCreateView realiza una llamada a la clase ReglasForm para conocer la estructura que debe seguir para la inserción de un nueva “Regla”.

Diagrama de Clases del Diseño

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software, así como sus relaciones proporcionan una perspectiva estática del sistema (representan su diseño estructural)

(Rumbaugh, 1999). A continuación, se muestra el Diagrama de Clases del Diseño de la funcionalidad Crear Punto en el requisito Gestionar Puntos:

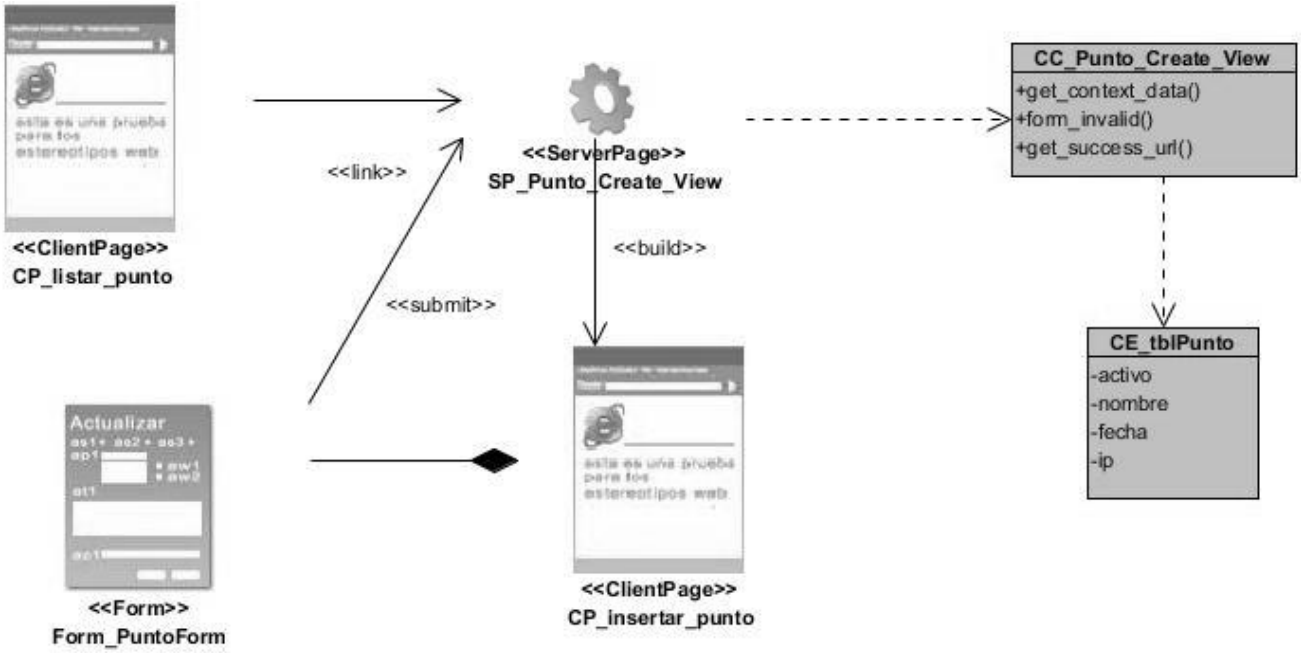


Ilustración 9: Diagrama de Clases del Diseño (Elaboración propia)

Modelo de datos

Si los requerimientos del software incluyen la necesidad de crear, ampliar o hacer interfaz con una base de datos, o si deben construirse y manipularse estructuras de datos complejas, el equipo del software tal vez elija crear un modelo de datos como parte del modelado general de los requerimientos. Un ingeniero o analista de software define todos los objetos de datos que se procesan dentro del sistema, la relación entre ellos y otro tipo de información que sea pertinente para las relaciones. El diagrama entidad-relación (DER) aborda dichos aspectos y representa todos los datos que se introducen, almacenan, transforman y generan dentro de una aplicación (Roger S. Pressman, 2010).

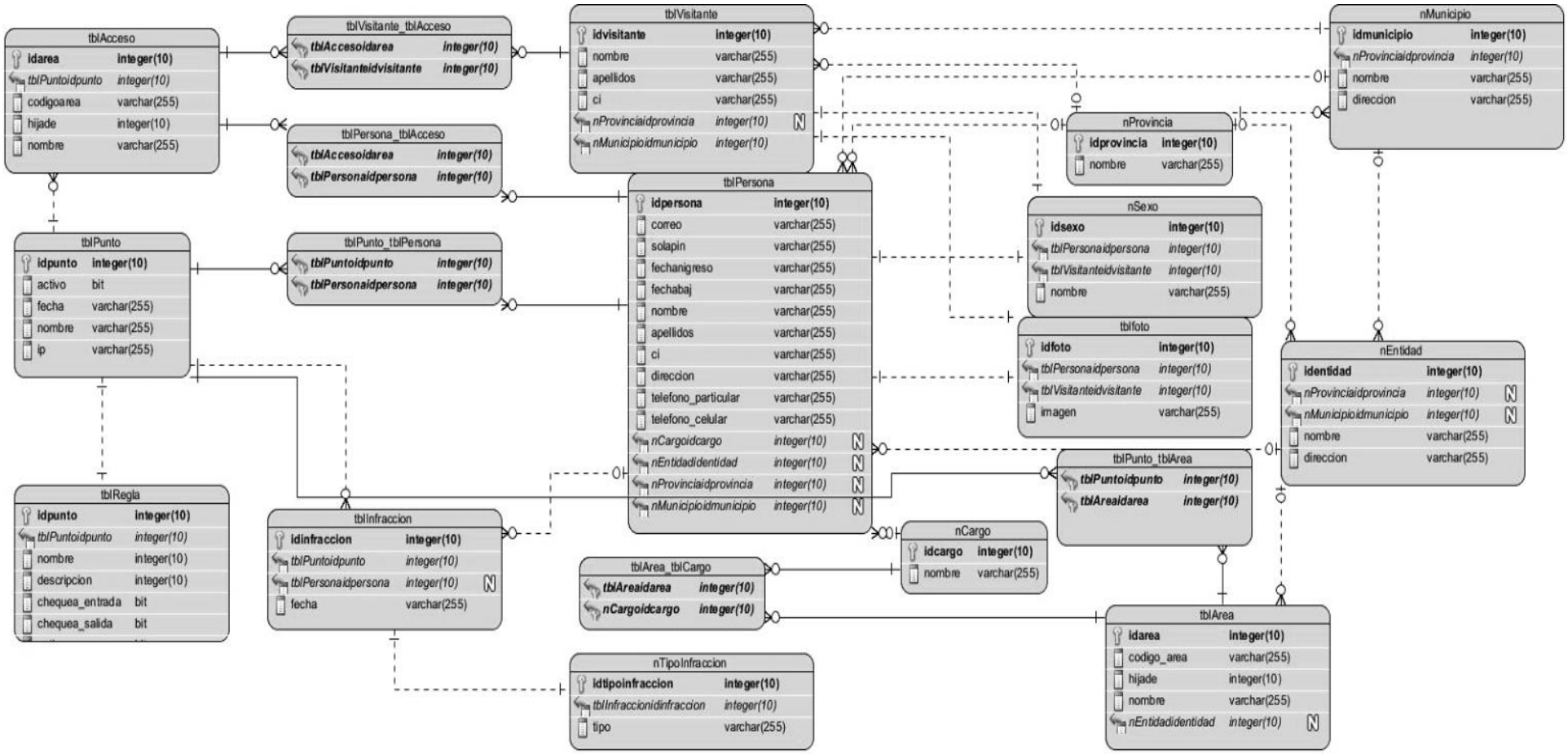


Ilustración 10: Diagrama Entidad-Relación (Elaboración Propia)

Diagrama de Despliegue

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos (Desarrollo de Software Orientado a Objeto usando UML., 2002).

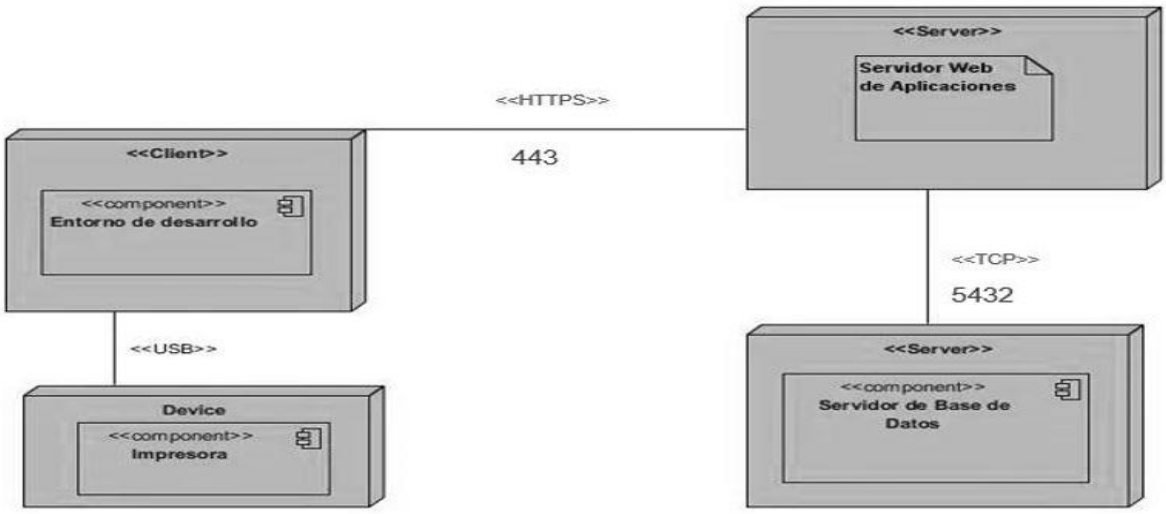


Ilustración 11: Diagrama de despliegue.

Descripción de elementos e interfaces de comunicación:

<<HTTPS>>: Hypertext Transfer Protocol Secure o HTTPS (en español protocolo de transferencia segura de hipertexto). Es un protocolo basado en el protocolo HTTP, establece a través del puerto 443 la conexión segura entre el dispositivo de acceso cliente y el servidor de aplicaciones. La conexión es por cable vía modem, Local Area Network (LAN) o red inalámbrica con una velocidad de más de 64 Kbps.

<<TCP/IP>>: estos protocolos establecen la conexión entre el servidor de aplicaciones y el servidor de base de datos. Para el servidor de base de datos de PostgresSQL se define el puerto 5432. La conexión entre el servidor web y el servidor de base de datos permite dar órdenes y obtener información de esta.

Conclusiones del capítulo:

Durante el desarrollo de este capítulo fueron definidas características esenciales para el análisis y diseño del módulo de control de acceso para el SIPP, lo que permitió arribar a las siguientes conclusiones:

- Mediante el análisis y especificación de los requisitos funcionales y no funcionales, así como la generación de las historias de usuario, se logró obtener una visión más clara de las funcionalidades a implementar y una mayor comprensión del proceso de negocio.
- La definición de una arquitectura para el desarrollo del software y la aplicación de los patrones de diseño permitió establecer las características de los componentes y el código fuente y establecer prácticas positivas en la fase de implementación.

Capítulo 3: Implementación y pruebas al módulo

En este capítulo quedarán definidos los resultados generados en las etapas de implementación y prueba, así como los estándares de codificación a seguir, el diagrama de componentes del módulo y las evidencias de las distintas pruebas realizadas al mismo.

Implementación

Una vez definidas las HU y concluido el diseño, corresponde la fase de implementación de la solución propuesta. Los objetivos de esta fase van destinados a desarrollar de forma iterativa e incremental un producto completo. Para una mayor comprensión se resume el funcionamiento del negocio en la siguiente ilustración:

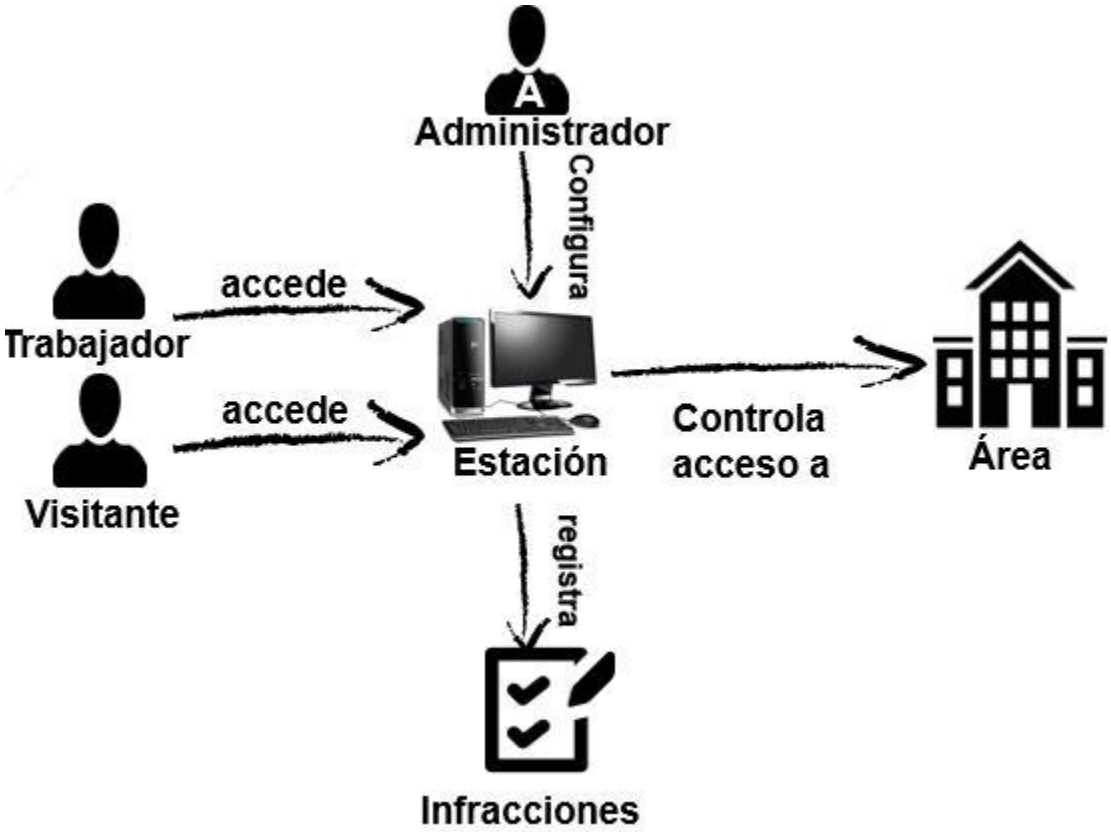


Ilustración 12: Explicación de la solución (Elaboración propia)

Estándares de codificación:

Sería lógico afirmar que el código es leído más veces de lo que es escrito, por lo que se hace necesario con el objetivo de lograr consistencia y coherencia en el código establecer pautas a seguir por los programadores que permitan mejorar la legibilidad del mismo, a partir de estas necesidades surgen los estándares o convenciones de código, los cuales definen un grupo de acuerdos para escribir código fuentes en la mayoría de los lenguajes de programación.

Los estándares de codificación acordados para el desarrollo de la presente solución son los definidos en el documento PEP 8—Style Guide for Python Code, por Guido van Rossum (Rossum, 2018). A continuación, quedan definidos:

Indentación:

- Las líneas de continuación deben alinearse verticalmente con el carácter que se ha utilizado (paréntesis, llaves, corchetes).
- Utilizar una indentación de una tabulación para cada línea con excepción de la primera.
- La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios.

Máxima longitud entre líneas:

- Todas las líneas deben estar limitadas a un máximo de setenta y nueve caracteres.
- Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas.
- En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas.

Líneas en blanco:

- Separar las funciones de alto nivel y definiciones de clases con dos líneas en blanco.
- Las definiciones de métodos dentro de una clase deben separarse por una línea en blanco.
- Se puede utilizar líneas en blanco escasamente para separar secciones lógicas.

Codificaciones:

- Utilizar la codificación UTF-8.
- Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U”.

Importaciones:

- Las importaciones deben estar en líneas separadas.
- Siempre deben colocarse al comienzo del archivo.

  Deben quedar agrupadas de la siguiente forma:

  1. Importaciones de la librería estándar.
  2. Importaciones terceras relacionadas.
  3. Importaciones locales de la aplicación/librerías.

- Cada grupo de importaciones debe estar separado por una línea en blanco.
- Evitar utilizar espacios en blanco en las siguientes situaciones:

    1. Inmediatamente dentro de paréntesis, corchetes y llaves.

    2. Inmediatamente antes de una coma, un punto y coma o dos puntos.

    3. Antes del paréntesis que comienza la lista de argumentos en la llamada a una función.

    4. Inmediatamente antes de un corchete que empieza una indexación.

    5. Más de un espacio alrededor de un operador de asignación (u otro) para alinearlo con otro.

Espacios en blancos en expresiones y sentencias:

- Deben rodearse con exactamente un espacio los siguientes operadores binarios:

    1. Asignación (=).

    2. Asignación de aumentación (+=, -=, etc.).

    3. Comparación (==, <, >,>=, <=, !=, <>, in, not in, is, is not).

    4. Expresiones lógicas (and, or, not).
- Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad.
- No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto.

Comentarios:

- Los comentarios deben ser oraciones completas.
- Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.
- Nunca cambiar las minúsculas y mayúsculas en los identificadores de clases, objetos, funciones, etc.
- Si un comentario es corto el punto final puede omitirse.

Cadenas de documentación:

- Deben quedar documentados todos los módulos, funciones, clases y métodos públicos.
- Para definir una cadena de documentación debe quedar encerrada dentro de (“”).
- Los (“”) que finalizan una cadena de documentación deben quedar en una línea a no ser que la cadena sea de una sola línea.

Convenciones de nombramiento:

- Nunca se deben utilizar como simple caracteres para nombres de variables los caracteres ele minúscula “l”, o mayúscula “O”, ele mayúscula “L” ya que en algunas fuentes son indistinguibles de los números uno y cero.
- Los módulos deben tener un nombre corto y en minúscula.
- Los nombres de clases deben utilizar la convención “CapWords” (palabras que comienzan con mayúsculas).
- Los nombres de las excepciones deben estar escrito también en la convención “CapWords” utilizando el sufijo “Error”.
- Los nombres de las funciones deben estar escrito en minúscula separando las palabras con un guion bajo “_”.
- Las constantes deben quedar escritas con letras mayúsculas separando las palabras por un guion bajo (_).

Diagrama de Componentes

El diseño de componentes para el software describe por completo los detalles internos de cada componente. Para lograrlo, este diseño define estructuras de datos para todos los objetos de datos locales y detalles algorítmicos para todo el procesamiento que tiene lugar dentro de un componente, así como la interfaz que permite el acceso a todas las operaciones de los componentes(comportamientos) (Roger S. Pressman, 2010). En la ilustración 18 se muestran los componentes y librerías del módulo de control de acceso:

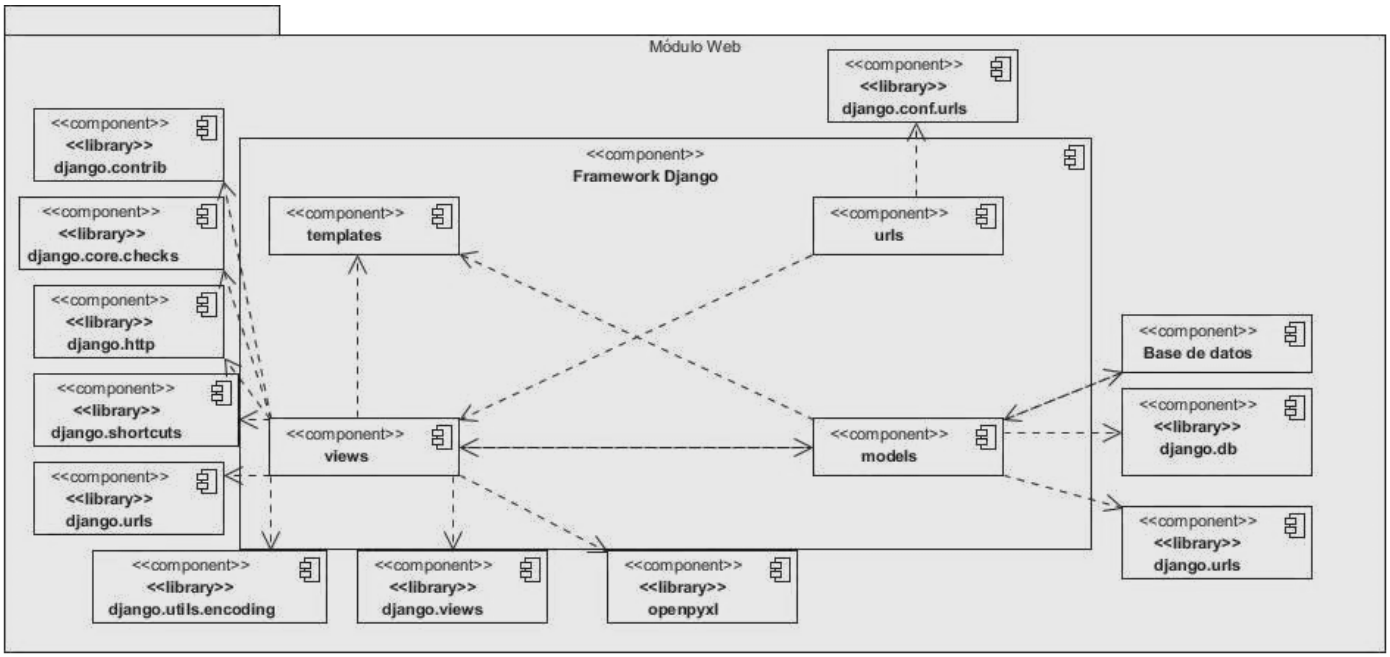


Ilustración 13: Diagrama de Componentes (Elaboración propia)

Pruebas

Las pruebas de una aplicación es una colección de actividades relacionadas con una sola meta: descubrir errores en el contenido, función, utilidad, navegabilidad, rendimiento, capacidad y seguridad de esa aplicación. Para lograr esto, se aplica una estrategia de prueba que abarca tanto revisiones como pruebas ejecutables (Roger S. Pressman, 2010).

La prueba de caja blanca del software se basa en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles (Roger S. Pressman, 2010).

La prueba de caja negra se refiere a las pruebas que se llevan a cabo en la interfaz del software. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software (Roger S. Pressman, 2010).

Pruebas de Rendimiento:

Las pruebas de rendimiento se usan para descubrir problemas de rendimiento que pueden ser resultado de: falta de recursos en el lado servidor, red con ancho de banda inadecuada, capacidades de base de

datos inadecuadas, capacidades de sistema operativo deficientes o débiles, funcionalidad de webapp pobremente diseñada y otros conflictos de hardware o software que pueden conducir a rendimiento cliente-servidor degradado (Roger S. Pressman, 2010).

Carga

La intención de la prueba de carga es determinar cómo responderán las webapps y su entorno del lado servidor a varias condiciones de carga (Roger S. Pressman, 2010).

Estrés

La prueba de esfuerzo o estrés es una continuación de la prueba de carga, pero en esta instancia las variables se fuerzan a satisfacerse y luego se superan los límites operativos (Roger S. Pressman, 2010).

Resultados de las pruebas de rendimiento

Con el objetivo de realizar las pruebas de rendimiento se utilizó la herramienta Apache Jmeter versión 3.2.

A continuación, se definen las propiedades de la PC utilizada como servidor y cliente:

Hardware PC (cliente y servidor):

Microprocesador: Intel(R) Core(TM) i-53210M CPU @ 2.50GHz

RAM: 4GB DDR3

Red: Ethernet 10/100Mbps

Software:

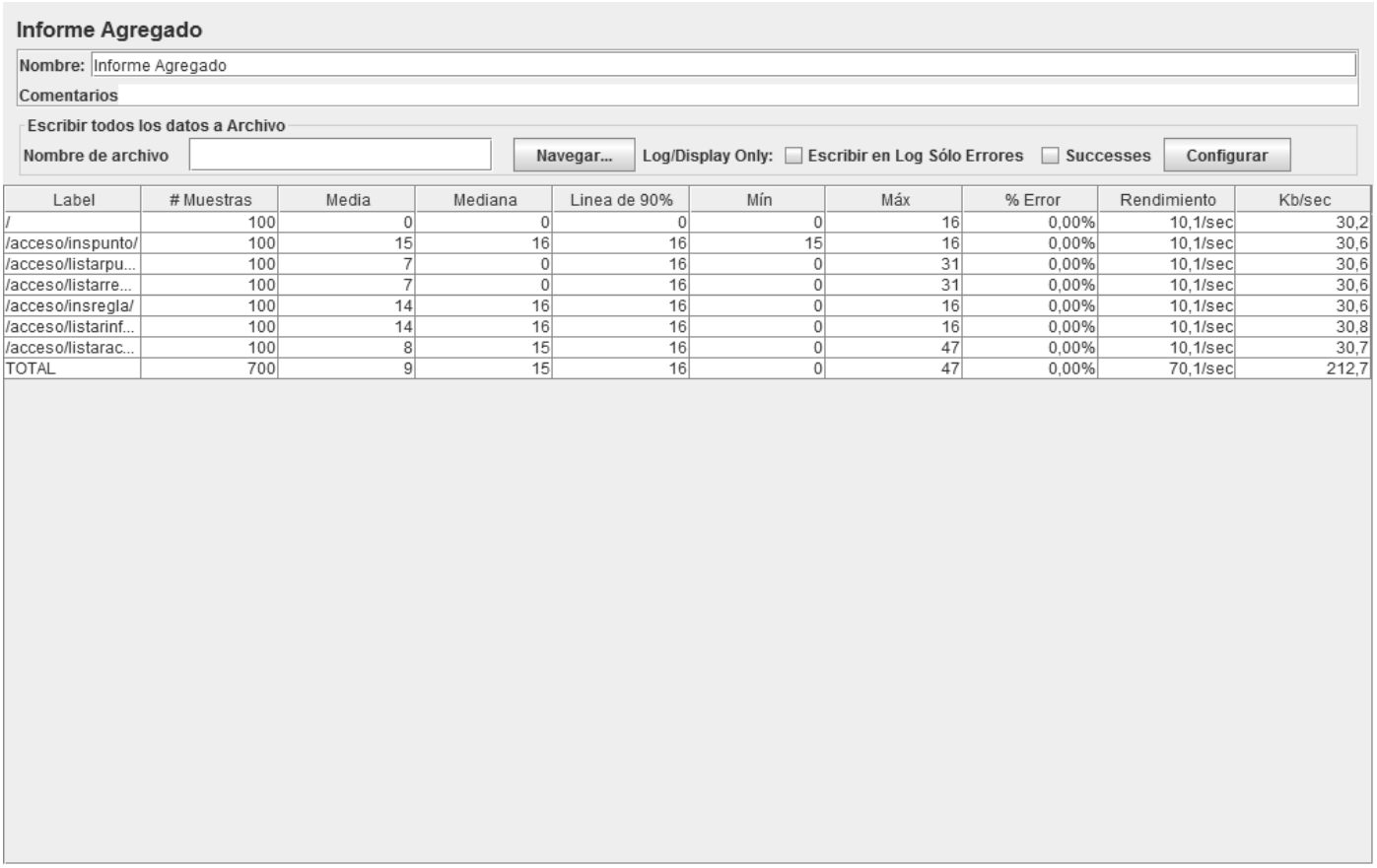
Plataforma: SO Windows

Servidor de Base de Datos: PostgreSQL 9.4

Luego de definido el Hardware se configuran los parámetros del Apache JMeter logrando un ambiente de simulación con un total de cien usuarios conectados concurrentemente. En la ilustración 19 se pueden observar los resultados obtenidos por el sistema. Para un mejor entendimiento de los componentes que se verán a continuación, se explica cada parámetro que la compone:

- **#Muestras:** cantidad de hilos utilizados para la URL.
- **Media:** tiempo promedio en milisegundos para un conjunto de resultados.

- **Min:** tiempo mínimo que demora un hilo en acceder a una página.
- **Max:** tiempo máximo que demora un hilo en acceder a una página.
- **Rendimiento**: rendimiento medido en los requerimientos por segundo / minuto / hora.
- **Kb/sec:** rendimiento medido en Kbytes por segundo.



Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/	100	0	0	0	0	16	0,00%	10,1/sec	30,2
/acceso/inspunto/	100	15	16	16	15	16	0,00%	10,1/sec	30,6
/acceso/listarpu...	100	7	0	16	0	31	0,00%	10,1/sec	30,6
/acceso/listarre...	100	7	0	16	0	31	0,00%	10,1/sec	30,6
/acceso/insregla/	100	14	16	16	0	16	0,00%	10,1/sec	30,6
/acceso/listarinf...	100	14	16	16	0	16	0,00%	10,1/sec	30,8
/acceso/listarac...	100	8	15	16	0	47	0,00%	10,1/sec	30,7
TOTAL	700	9	15	16	0	47	0,00%	70,1/sec	212,7

Ilustración 14: Resultado de las pruebas de carga y estrés.

Análisis de los resultados de las pruebas de rendimiento

El tiempo promedio de las solicitudes es de 9 milisegundos, realizándose 700 solicitudes al servidor con una frecuencia de 10 segundos, obteniéndose un 0% de error en todas las peticiones realizadas.

Pruebas Funcionales

Estas pruebas se realizan con el objetivo de descubrir errores que indican falta de conformidad con los requerimientos del cliente. Cada función de la aplicación se valora en su corrección, inestabilidad y conformidad con los estándares de implantación adecuados (Roger S. Pressman, 2010). En las siguientes tablas se muestran las pruebas realizadas al requisito Gestionar Reglas (ver Anexo 3).

Tabla 2: Variables para los casos de prueba funcional gestionar Reglas (Elaboración propia)

No.	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Nombre	Campo de texto	No	Se introduce el nombre de la regla.	NR
2	Descripción.	Campo de texto	No	Se realiza una breve descripción de la regla.	D
3	Entrada	Checkbox	Solo si se seleccionó salida.	Se selecciona si la regla permite controlar la entrada.	E
4	Salida	Checkbox	Solo si se seleccionó entrada.	Se selecciona si la regla permite controlar salida.	S
5	Activa	Checkbox	Si	Se selecciona si la regla estará activa en el momento de su creación	A

Tabla 3: Caso de prueba funcional Insertar Regla ESC 1 (Elaboración propia)

Descripción general								
Permitirá insertar una nueva regla.								
Condiciones de ejecución								
Para insertar una nueva regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	NR	D	**E**	**S**	**A**	**R/ del sistema**	**Flujo central**
ESC 1.1 Introducir datos de la regla correctamente**.**	El usuario llena y activa todos los campos	Nueva	Nueva Regla	Si	Si	Si	El sistema crea una regla satisfactoriamente y muestra	En el menú a la izquierda se sigue la siguiente ruta:

	del formulario.						notificación de éxito.	“Control de Acceso - Administración Reglas- Nueva". Aparecerá un formulario con los campos a llenar.

Tabla 4: Caso de prueba funcional Modificar Regla ESC 1 (Elaboración propia)

Descripción general								
Permitirá modificar una regla.								
Condiciones de ejecución								
Para modificar una nueva regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	NR	D	**E**	**S**	**A**	**R/ del sistema**	**Flujo central**
ESC 1.2 Modificar datos de la Regla correctamente**.**	El usuario selecciona el botón modificar.	Nueva1	Nueva Regla	Si	Si	Si	El sistema modifica la Regla satisfactoriamente y muestra notificación de éxito.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Reglas- Modificar.

Tabla 5: Caso de prueba funcional Eliminar Regla ESC 1 (Elaboración propia)

Descripción general								
Permitirá eliminar una regla.								
Condiciones de ejecución								
Para eliminar regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios. Debe existir alguna regla creada								
Escenario	**Descripción**	NR	D	**E**	**S**	**A**	**R/ del sistema**	**Flujo central**
ESC 1.3 Eliminar la regla correctamente**.**	El usuario selecciona el botón eliminar.	N/S	N/S	N/S	N/S	N/S	El sistema elimina la regla satisfactoriamente y muestra notificación de éxito.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Reglas- Eliminar.

Tabla 6: Caso de prueba funcional Insertar Regla ESC 2 (Elaboración propia)

Descripción general								
Permitirá insertar una nueva regla.								
Condiciones de ejecución								
Para insertar una nueva regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	NR	D	**E**	**S**	**A**	**R/ del sistema**	**Flujo central**
ESC 2.1 Introducir datos de la regla vacíos**.**	El usuario no llena los campos del formulario.	vacío	vacío	vacío	vacío	vacío	El sistema cancela la inserción de la regla y muestra un mensaje de error “Este campo es necesario”, para los campos nombre y descripción.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Reglas-Nueva". Aparecerá un formulario con los campos a llenar.

Tabla 7: Caso de prueba funcional Modificar Regla ESC 2 (Elaboración propia)

Descripción general								
Permitirá modificar una regla.								
Condiciones de ejecución								
Para modificar una nueva regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	NR	D	**E**	**S**	**A**	**R/ del sistema**	**Flujo central**
ESC 2.2 Introducir datos de la regla vacíos**.**	El usuario selecciona el botón modificar e intenta introducir los datos de la regla vacíos.	vacío	vacío	vacío	vacío	vacío	El sistema cancela la inserción de la regla y muestra un mensaje de error “Este campo es necesario”, para los campos nombre y descripción.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Reglas- Modificar.

Tabla 8: Caso de prueba funcional Insertar Regla ESC 3 (Elaboración propia)

Descripción general								
Permitirá insertar una regla.								
Condiciones de ejecución								
Para insertar una nueva regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	NR	D	**E**	**S**	**A**	**R/ del sistema**	**Flujo central**
ESC 3.1 Introducir datos de la regla no válidos**.**	El usuario introduce caracteres numéricos en el campo Nombre	34141	Nueva Regla	Si	Si	Si	El sistema cancela la inserción de la regla y muestra un mensaje de error “Caracteres no válidos”, para el campo nombre.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Reglas- Nueva". Aparecerá un formulario con

Descripción general								
Permitirá modificar una regla.								
Condiciones de ejecución								
Para modificar una regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	NR	D	**E**	**S**	**A**	**R/ del sistema**	**Flujo central**
ESC 3.2 Introducir datos de la regla no válidos**.**	El usuario introduce caracteres numéricos en el campo Nombre	34141	Nueva Regla	Si	Si	Si	El sistema cancela la inserción de la regla y muestra un mensaje de error “Caracteres no válidos”, para el campo nombre.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Reglas- Modificar". Aparecerá un formulario con los campos a llenar.
								los campos a llenar.

Tabla 9: Caso de prueba funcional Modificar Regla ESC 3 (Elaboración propia)

Resultados de las pruebas funcionales

Con el objetivo de validar los requisitos funcionales se realizaron tres iteraciones por requisito en las que se encontraron 12 no conformidades después de haber corregido las mismas, se realizó una segunda iteración en la que se encontraron 5 no conformidades quedando resueltas para la tercera iteración en la que no se encontraron no conformidades. En la ilustración 20 se muestran de manera gráfica los resultados obtenidos.

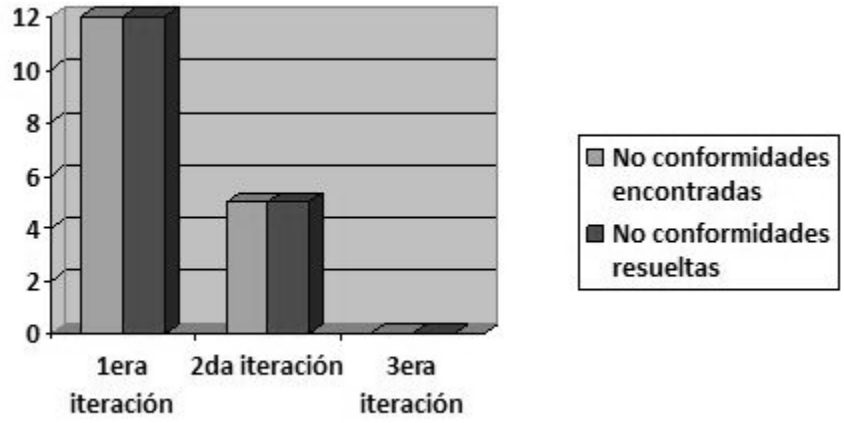


Ilustración 15: No conformidades por iteración (Elaboración propia)

Las no conformidades encontradas fueron:

- Introducción de caracteres no válidos en campos de textos.
- Algunos mensajes de error muestran los nombres de atributos.
- Errores ortográficos en los nombres de los campos de formularios.

Prueba de Integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño (Roger S. Pressman, 2010).

Para verificar la integración del Módulo de Control de Acceso para el SIPP con los demás módulos que conformaran la solución se realizó la integración con el Módulo de Administración para el SIPP quedando verificado que no se ve afectado el funcionamiento de la aplicación y pudiendo ser utilizados los roles y

permisos definidos por el Módulo de Administración para controlar el acceso a las funcionalidades del Módulo de Control de Acceso. En la siguiente ilustración queda evidenciado el uso de los permisos:

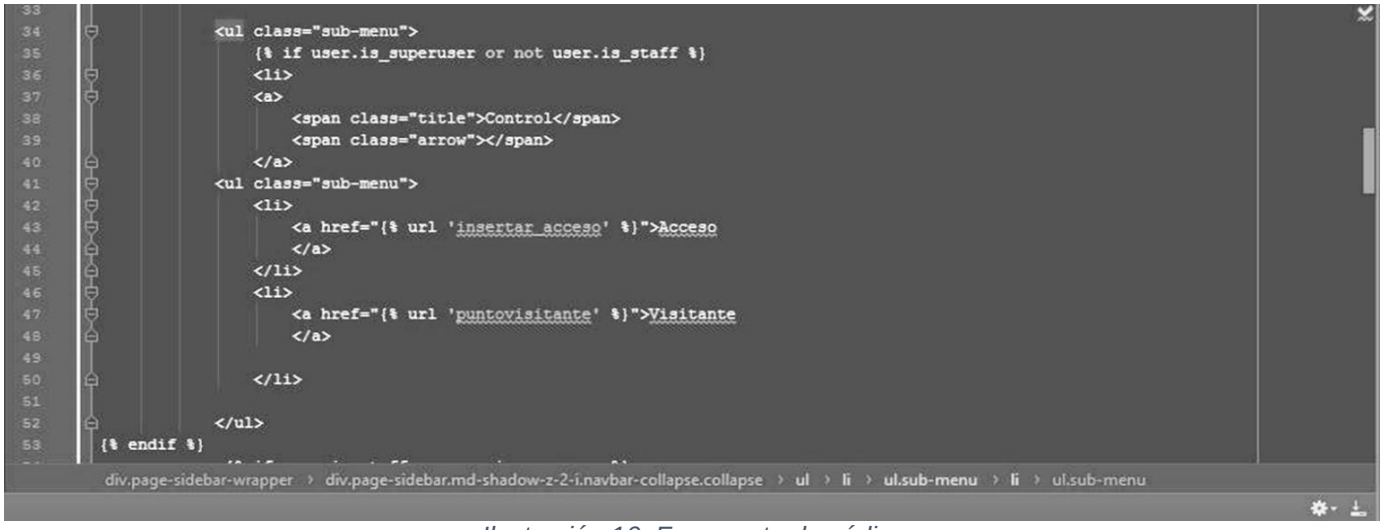


Ilustración 16: Fragmento de código.

Una vez verificado que el conjunto funciona de acuerdo con lo previsto, fue sumado un nuevo módulo. Luego del acoplamiento de cada uno, se efectuaron pruebas para comprobar la correcta interacción entre los mismos. Este proceso fue realizado para todos los componentes del software. Durante la integración no fueron detectados nuevos errores y la comunicación entre los módulos fue exitosa.

Prueba de Seguridad

La prueba de seguridad intenta verificar que los mecanismos de protección que se construyen en un sistema en realidad lo protegerán de cualquier penetración impropia (Roger S. Pressman, 2010).

Para la realización de la prueba de seguridad se utilizó la herramienta Acunetix basadas principalmente en:

- Inyección HTTP
- Cross-Site Scripting
- Falsificación de petición
- Inyección SQL

Después de haber realizado una primera y segunda iteración se detectaron vulnerabilidades de los siguientes tipos:

- Debilidades en casos de ataques de fuerza bruta al Login.
- Credenciales en texto plano
- Posibles virtual hosts

Todas las vulnerabilidades fueron tomadas en cuenta y se realizaron cambios en la solución, quedando resueltos los problemas como se pudo comprobar en una tercera iteración de la prueba, a continuación, se muestran los resultados de las tres iteraciones:

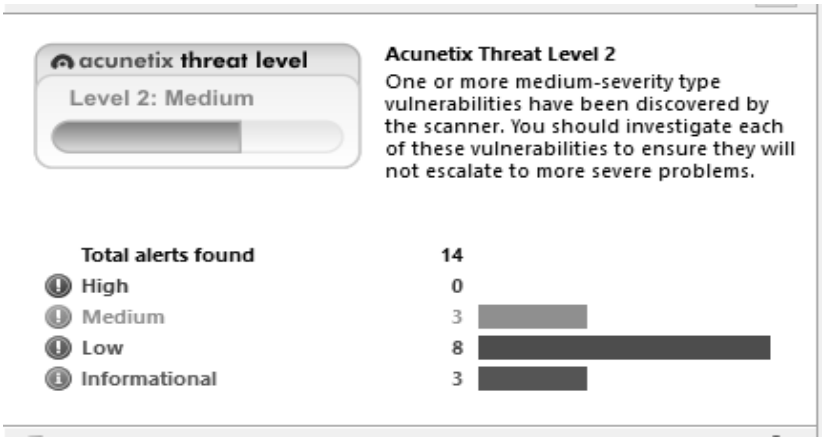


Ilustración 17: Prueba de Seguridad primera iteración.

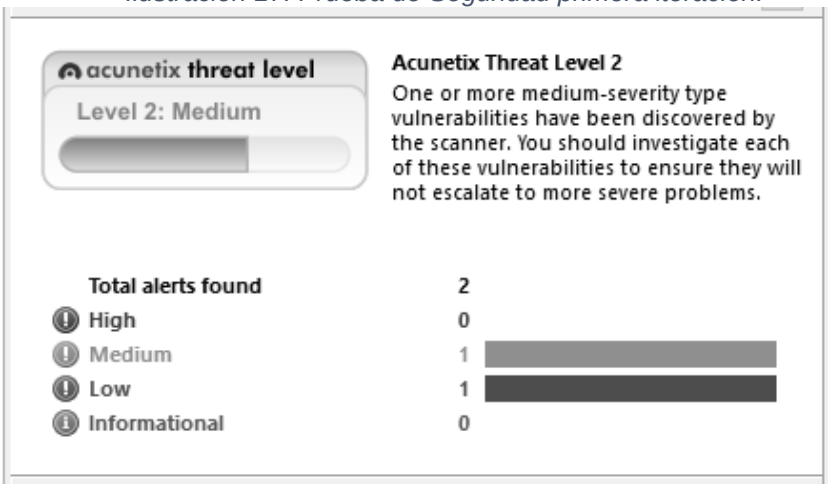


Ilustración 18: Prueba de Seguridad segunda iteración.

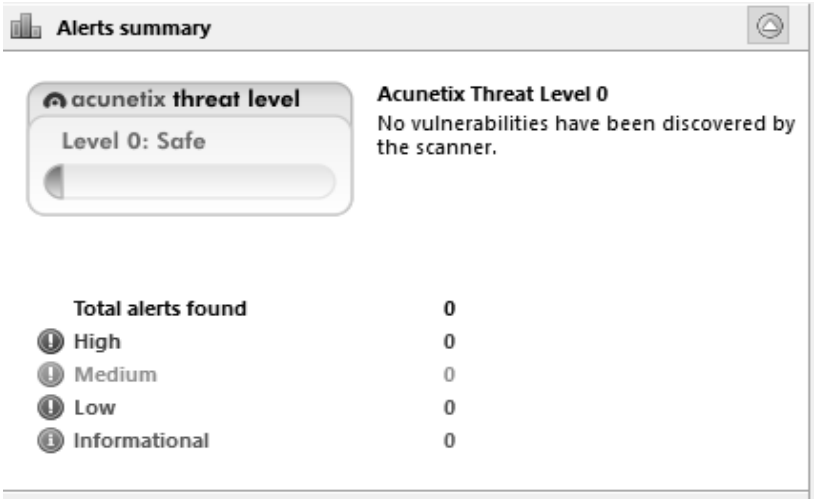


Ilustración 19: Prueba de Seguridad tercera iteración.

Validación de la Hipótesis:

Con el objetivo de validar la hipótesis de la investigación se aplica el método de consulta a expertos, quedando definido en los siguientes puntos a seguir para la validación:

- Definición de las características de selección de expertos.
- Identificación y valoración de posibles expertos.
- Selección de expertos.
- Implementación de las consultas a expertos, procesamiento y evaluación de los resultados obtenidos.

Definición de las características de selección de expertos:

Para la identificación de posibles expertos se tendrá en cuenta las siguientes características:

- Disposición a participar en la encuesta.
- Competencia.
- Conocimiento de tecnologías web.
- Relación con el tipo de módulo desarrollado.
- Experiencia laboral.

Expertos identificados y seleccionados, evaluación de las características de selección

Tabla 10: Expertos utilizados en la validación de la propuesta de solución.

No.	Experto	Entidad	Años de experiencia	Conocimientos Web
1	Osay Gonzáles Fuentes	CISED	6	Si
2	José Carlos Pérez Zamora	CISED	5	Si
3	Miguel Jaeger Rodríguez Lazo	FAC-1 Programación	7	Si
4	Ernesto Soto Gómez	FAC-1 ISW	5	Si
5	Denier Naranjo Oliva	CISED	5	Si

Luego de realizar la selección de los expertos se generó un instrumento que permitirá validar el módulo de control de acceso para el SIPP. El instrumento se encuentra compuesto de 5 preguntas relacionadas al funcionamiento del módulo. Los expertos podrán expresar sus valoraciones en las siguientes categorías:

- Muy adecuado (MA).
- Adecuado (A).
- Poco adecuado (PA).
- Inadecuado (I).

Tabla 11: Sentencias a evaluar por los expertos para validar la hipótesis científica.

No.	Sentencias plasmadas en la consulta realizada a expertos
1	El módulo realiza el proceso de control de acceso correctamente.
2	El módulo permite conocer el flujo de acceso a las distintas áreas de la entidad.
3	El módulo posibilita asegurar la información de una entidad.
4	El módulo es capaz de generar reportes de acceso e infracciones.
5	El módulo brinda opciones para la gestión de visitantes a una entidad.

Una vez realizada la encuesta a los expertos seleccionados se obtuvo como resultado (ver Anexo 4), lo que permite llegar a las siguientes conclusiones:

- El 100% de los expertos está coincide que el módulo:
    - ✓ realiza el proceso de control de acceso correctamente.
    - ✓ permite conocer el flujo de acceso a las distintas áreas de la entidad.
    - ✓ es capaz de generar reportes de acceso e infracciones.
    - ✓ brinda opciones para la gestión de visitantes a una entidad.
- El 60% de los expertos considera muy adecuada la afirmación de que el módulo posibilita asegurar la información de una entidad, el restante 40% considera la afirmación adecuada.

Los resultados obtenidos demuestran la aceptación de los expertos con respecto a la solución implementada valorando sus principales funcionalidades en muy adecuadas y adecuadas, y de esta forma se demuestra la validez de la hipótesis científica de la investigación, evidenciándose que la implementación del módulo de control de acceso y registro de visitantes para el SIPP permitirá garantizar la seguridad de la información y conocer el flujo de acceso a las distintas áreas de la entidad.

Conclusiones del capítulo

Después de haber desarrollado el presente capítulo se arribó a las siguientes conclusiones:

- La definición de un estándar de codificación permitió aumentar la claridad y reutilización del código, así como establecer pautas para una mejor comprensión del mismo.
- El desarrollo de un diagrama de componentes permitió una mayor comprensión estructural de la solución.
- Con la verificación de la solución basándose en un desarrollo guiado por pruebas se mitigaron las no conformidades, asegurando así la funcionalidad, seguridad y rendimiento del módulo.

Conclusiones:

Finalizada la investigación se arriba a las siguientes conclusiones:

- El estudio del marco teórico permitió constatar los conceptos fundamentales en cuanto a sistemas de control de acceso y características esenciales de los mismos.
- El análisis del estudio del estado del arte posibilitó establecer que los sistemas de control de acceso estudiados no constituyen soluciones capaces de resolver la problemática planteada en la investigación debido sus altos costos en conceptos de hardware *y* licencias, e identificar características funcionales a implementar en la solución.
- La modelación de los artefactos posibilitó establecer las características de los componentes, la organización lógica del código fuente y establecer prácticas positivas en la fase de implementación.
- El Módulo de Control de Acceso y Registro de Visitantes constituye una solución funcional y con calidad, conforme a los resultados obtenidos de las pruebas funcionales, rendimiento, seguridad e integración.
- La implementación Módulo de Control de Acceso y Registro de Visitantes permitió lograr la seguridad de la información y el control del flujo de acceso a las distintas áreas de la entidad y proporciona una solución aceptable a la situación problemática existente.

Recomendaciones:

- A los especialistas del Centro CISED, valorar la posibilidad de incorporar funcionalidades de lectores biométricos para registrar los datos en el proceso de control de acceso.
- A los analistas del Centro CISED, elaborar un manual de usuario de la solución con el objetivo de apoyar el aprendizaje del personal que utilizará el SIPP.

Bibliografía

[En línea] [Citado el: 25 de 11 de 2017.] http://www.protection1.com/business/security-access-control-systems/.

[En línea] [Citado el: 26 de 11 de 2017.] http://www.datys.cu/spa/site/product/20.

An access control model for cloud computing. **YA, Tounis. 2014.** 2014, Vol. 19.

asmag. [En línea] [Citado el: 22 de 11 de 2017.] https://www.asmag.com/showpost/22408.aspx.

Baechli, Kathleen Jui. 2005. *Comparación de las tecnologías de control de acceso a las instalaciones en una organización.* 2005.

Baryolo, Oinier Gómez. 2012. *CAEM:Modelo de Control de Acceso para Sistemas de Información en Entornos Multidominios.* La Habana : s.n., 2012.

Beoto, Elizabeth Santana. 2010. *COSMO. Sistema de Gestión de Laboratorios o Agrupaciones de Computadoras. Tesis presentada en opción al título estatal de Máster en Ciencias. .* La Habana : s.n., 2010.

Cadavid, An. 2013. *Revisión de metodologías ágiles para el desarrollo de software.* s.l. : Prospectiva, 2013.

D.Ferraiolo. 2013. semanticsholar. [En línea] 2013. [Citado el: 21 de 11 de 2017.] http://www.semantischolarg.org/pdfs.

datys. [En línea] [Citado el: 25 de 11 de 2017.] http://www.datys.cu/spa/site/product/10.

Denzer, Patricio. 2002. [En línea] 23 de 10 de 2002. [Citado el: 28 de 11 de 2017.] http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf.

Desarrollo de Software Orientado a Objeto usando UML. **Letelier, Torres Patricio. 2002.** Valencia : s.n., 2002.

2014. dte. [En línea] 13 de 1 de 2014. [Citado el: 21 de 11 de 2017.] www.dte.us.es/docencia/etsii/gii-ti/isi/laboratorios/Lab4-Parte2-ACL.pdf.

Edeki, C. 2013. Agile Unified Process. *International Journal of Computer Science.* 2013.

Escalona, Darlin Díaz. 2016. *Gestion de Visitantes para la Plataforma Modular de Identificación y Control de Acceso.* La Habana : s.n., 2016.

Espinosa, Alejandro. 2017. ipusergrouplatino. [En línea] 21 de 3 de 2017. [Citado el: 22 de 11 de 2017.] http://ipusergrouplatino.com/articles/article/8391448/180360.

Etchart, Graciela. 2011. [En línea] 2011. [Citado el: 24 de 11 de 2017.] http://sedici.unlp.edu.ar/bitsstream/handle/10915/20052/Documento completo.pdf?sequence=1.

eweek. [En línea] [Citado el: 28 de 11 de 2017.] 35. http://www.eweek.com/development/jetbrains-strikes-python-developers-with-pycharm-1.0-ide.

Fernández, Luis Gómez. 2012. *Guía de aplicación de la Norma UNE-ISO/IEC 27001 sobre seguridad en sistemas de información para pymes.* Madrid : AENOR (Asociación Española de Normalización y Certificación), 2012.

Fowler, Martin. 1999. *UML Gota a Gota.* 1999.

Gómez, O Tinoco. 2010. *Criterios de selección de metodologías de desarrollo de software.* s.l. : Industrial Data, 2010.

honeywell. [En línea] [Citado el: 25 de 11 de 2017.] www.security.honeywell.com/es/productos/access/software/389765.html.

honeywellintegrated. [En línea] [Citado el: 25 de 11 de 2017.] https://www.honeywellintegrated.com/products/integrated-security/sms/543964.html.

J., Yang-Feng. 2013. *Acces Control for rural medical and health collaborative working plataform.* s.l. : The Journal of Chine Universities of Posts and Telecommunications , 2013.

Kim, Solomon. 2010. *Foundamentals of Information Systems Security.* s.l. : Jones & Bartlett Learning, 2010.

L.Drake. 2009. python. [En línea] 9 de 2009. [Citado el: 28 de 11 de 2017.] http://python.org.ar/pyar/Tutorial .

Larman, Craig. 2004. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2004.

Letelier, Patricio. 2003. *Actas Metodologías Ágiles en el desarrollo de Software.* Alicante : s.n., 2003.

Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos. **Hernández, P. 2017.** s.l. : Revista Antioqueña de las ciencias computacionales, 2017.

Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). **Letelier, Patricio. 2006.** 26, Buenos Aires : s.n., 2006, Vol. 05.

mozilla. [En línea] [Citado el: 28 de 11 de 2017.] 34. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction.

Ouellete, Jason. 2014. securitymagazine. [En línea] 9 de 12 de 2014. [Citado el: 23 de 11 de 2017.] https://www.securitymagazine.com/articles/85970-access-control-whats-on-the-horizon.

Patrón Modelo-Vista-Controlador. **Díaz Gonzáles, Yanette. 2012.** 1, La Habana : s.n., 2012, Vol. 11.

Patrones Grasp y Anti-Patrones: un enfoque Orientado a Objetos desde Lógica de Programación. **Botero Tabares, Ricardo. 2010.** No. 8, s.l. : Entre Ciencia e Ingeniería, 2010.

Roger S. Pressman, Ph.D. 2010. *Ingeniería del software, Un enfoque práctico.* Connecticut : Mcgraw Hill, 2010.

Rossum, Guido van. 2018. python.org. [En línea] 5 de 4 de 2018. [Citado el: 4 de 5 de 2018.] https://www.python.org/dev/peps/pep-0008/.

Rumbaugh, Jim. 1999. *El Lenguaje Unificado de Modelado .* s.l. : AddIson-Wesley, 1999.

2007. scribd. [En línea] 2007. [Citado el: 20 de 11 de 2017.] https://es.scribd.com/document/192308599/Resolucion-No-127-Del-2007-MIC.

sdmmag. [En línea] [Citado el: 25 de 11 de 2017.] https://www.sdmmag.com/articles/91757-brivo-launches-brivo-mobile-pass-to-open-doors-using-smartphones.

Sistema de control de acceso e interbloqueo para el Centro de Inmunología Molecular. **Pedreira, Ing. Marcel. 2013.** 3, La Haban : EAC, 2013, Vol. 34.

tarjenova. [En línea] [Citado el: 24 de 11 de 2017.] http://www.tarjenova.com/tarjetas-plasticas/tecnologia-de-lectura/codigo-barras/.

tyco. [En línea] [Citado el: 25 de 11 de 2017.] https://tycoifs.com.ar/comopodemos-ayudar/proteja-su-negocio/control-de-acceso/sistema-de-control-de-acceso.

uc3m. [En línea] [Citado el: 29 de 11 de 2017.] http://www.ie.inf.uc3m.es/grupo/docencia/reglada/Is1y2/PracticaVP.pdf.

uci. [En línea] [Citado el: 25 de 11 de 2017.] https://www.uci.cu/investigacion-y-desarrollo/productos.

Vignaga, Andrés. 2003. *Enfoque metodológico para el desarrollo basado en componentes.* Chile : s.n., 2003.

visual-paradigm. [En línea] [Citado el: 29 de 11 de 2017.] https://www.visual-paradigm.com/support/faq.jsp.

Anexo 1-Tecnologías y herramientas de control de acceso (ilustraciones).

Ilustración 20: Tarjeta de proximidad.

Ilustración 21: Lector de tarjeta de proximidad.

Ilustración 22: Tarjeta de código de barras.

Ilustración 23: Lector de tarjetas de código de barras.

Ilustración 24: Llave electrónica.

Ilustración 25: Lector de llaves electrónicas.

Anexo 2- Historias de usuario.

Tabla 12: HU-2 Gestionar Reglas.

Número: HU-2	**Nombre de requisito: Gestionar reglas**
Programador: Christian M. Paneque	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo:** 48 horas
Riesgo: Alto	**Tiempo real:** 32 horas
Descripción: **El usuario con los permisos adecuados será capaz de crear, modificar o eliminar una nueva regla para un punto, las reglas definen las características del punto, de las reglas se podrán definir: nombre, descripción, si el punto controlará entrada, salida o ambas y si la regla estará o no activa.**	
Observaciones: Según los permisos del usuario tendrá diferentes capacidad de edición.	

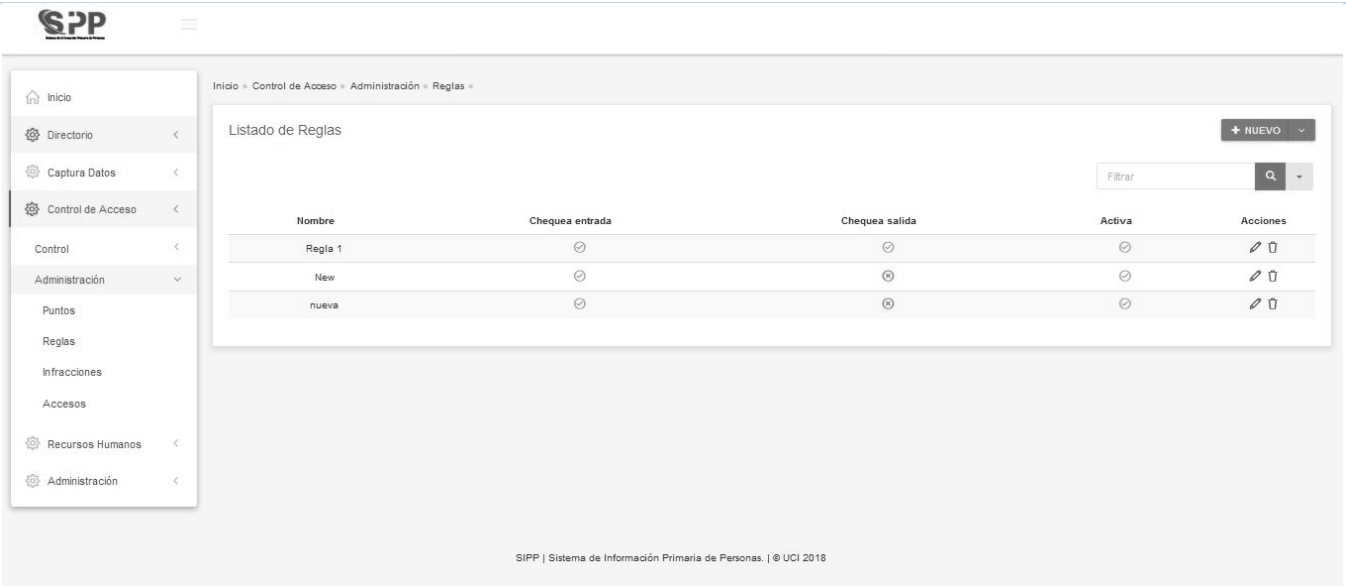


Tabla 13: HU-3 Controlar Acceso.

Número: HU-3	**Nombre de requisito: Controlar Acceso**
Programador: Christian M. Paneque	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo:** 72 horas
Riesgo: Alto	**Tiempo real:** 72 horas

Descripción:

El usuario con los permisos adecuados será capaz de, mediante el carnet de identidad de una persona o su solapín verificar si tiene acceso al área a la que controla.

Observaciones: Si el acceso es permitido se registrará un acceso si no lo es se registrará una infracción.

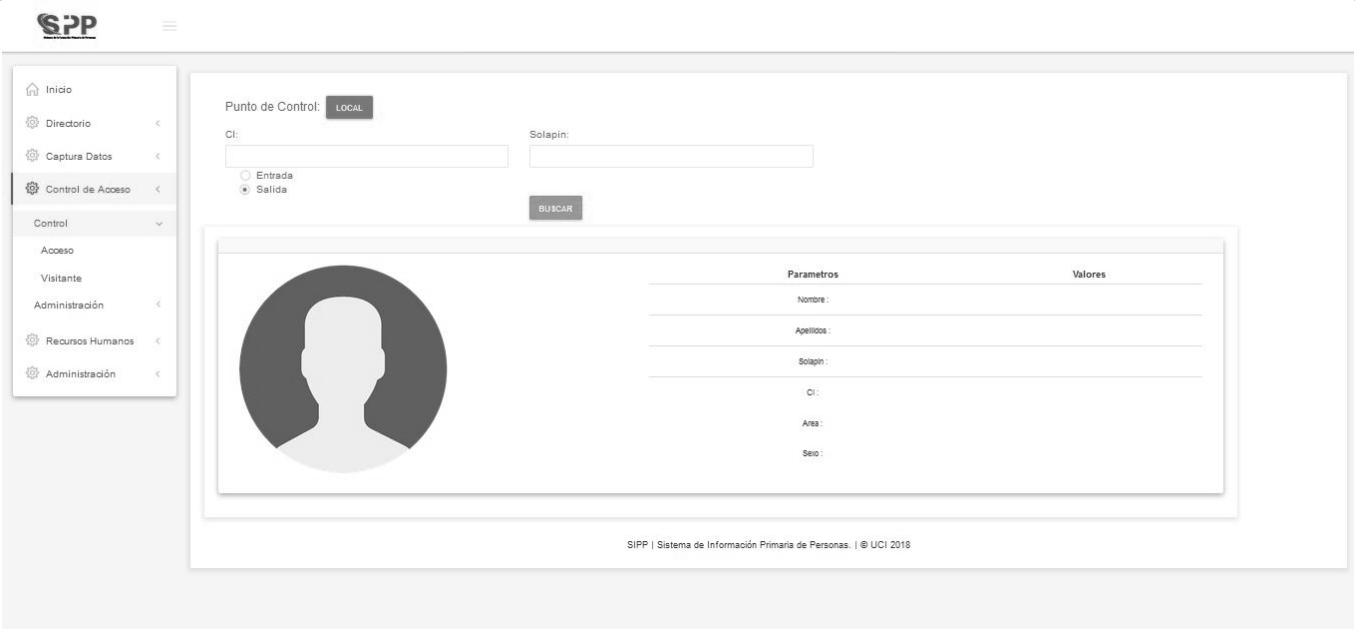


Tabla 14: HU-4 Insertar Visitante.

Número: HU-4	**Nombre de requisito: Insertar Visitante**
Programador: Christian M. Paneque	**Iteración Asignada:** 1
Prioridad: Media	**Tiempo:** 32 horas
Riesgo: Medio	**Tiempo real:** 24 horas
Descripción: **El usuario con los permisos adecuados será capaz de verificar si un visitante posee un solapín de visitante que le permita el acceso al área, también tendrá la capacidad de registrar un nuevo visitante y asignar un solapín de visitante al mismo**	
Observaciones: Los permisos de acceso de visitantes no son gestionados por la aplicación.	

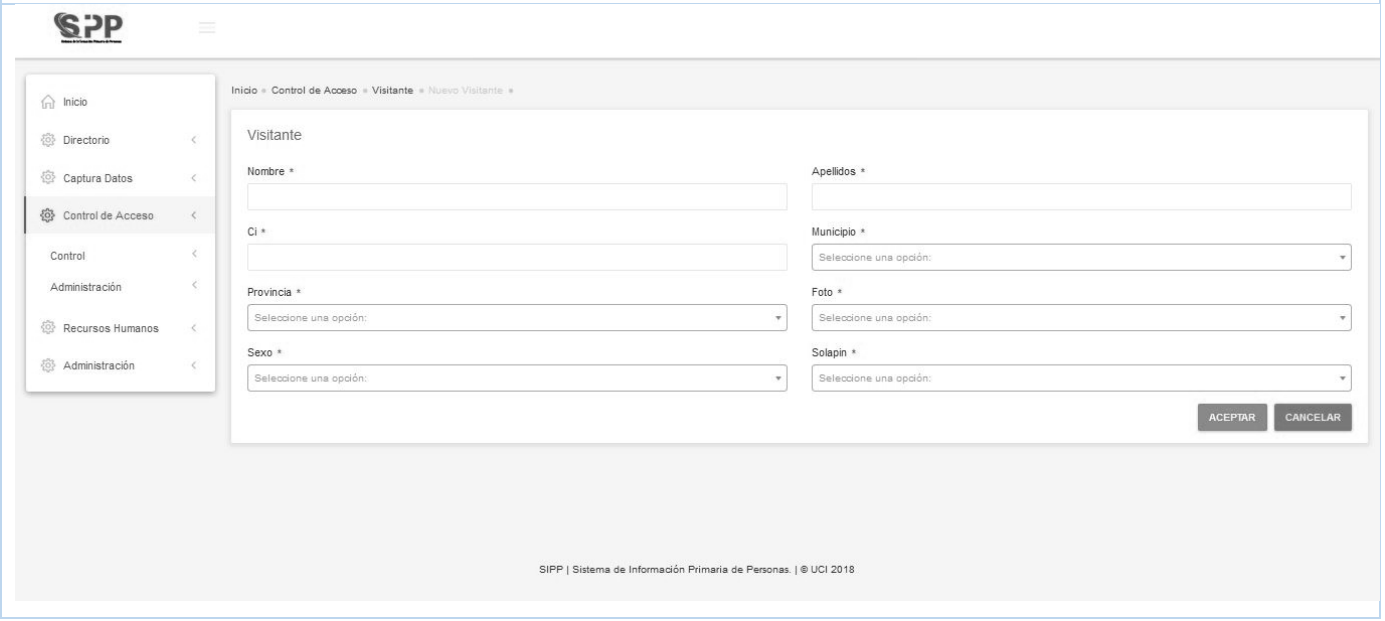


Tabla 15: HU-5 Listar Infracciones.

Número: HU-5	**Nombre de requisito: Listar Infracciones**
Programador: Christian M. Paneque	**Iteración Asignada:** 1
Prioridad: Media	**Tiempo:** 32
Riesgo: Medio	**Tiempo real:** 24
Descripción:	

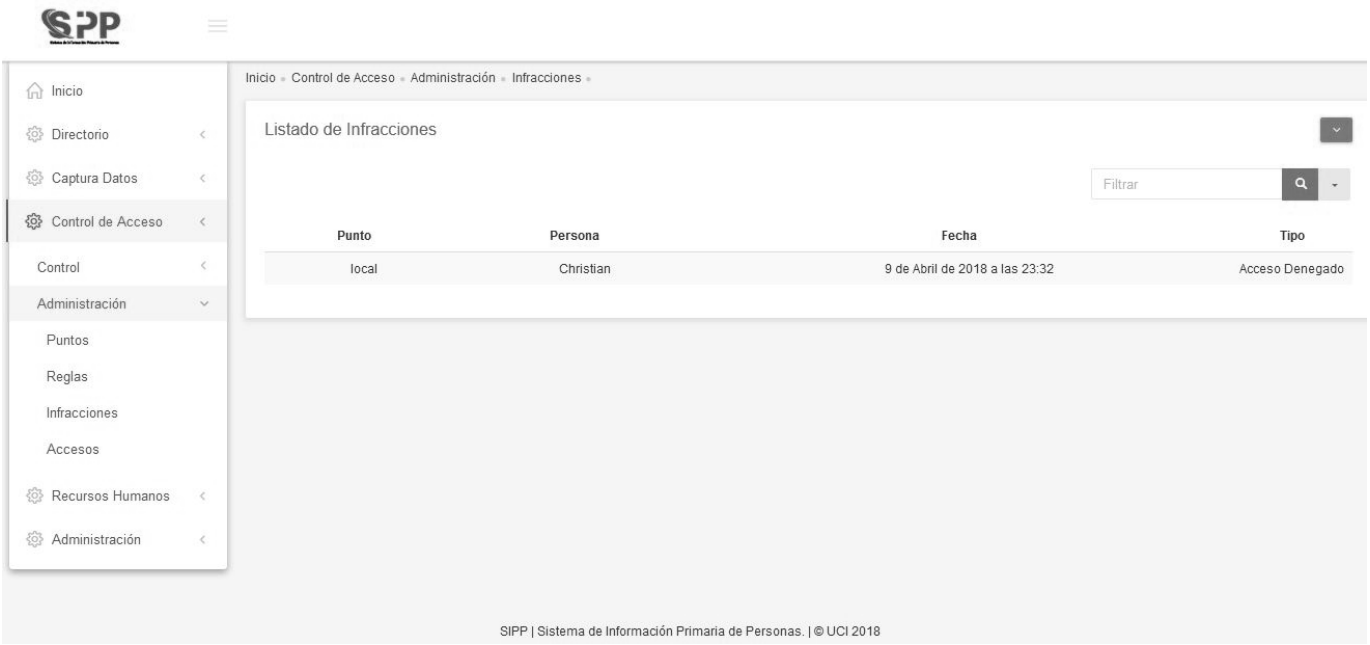
El usuario con los permisos adecuados será capaz de listar las infracciones ocurridas durante el proceso de acceso
Observaciones: Las Infracciones no podrán ser eliminadas o modificadas.


Tabla 16: HU-6 Listar Accesos.

Número: HU-6	**Nombre de requisito: Listar Accesos**
Programador: Christian M. Paneque	**Iteración Asignada:** 1
Prioridad: Media	**Tiempo:** 32
Riesgo: Medio	**Tiempo real:** 24
Descripción: **El usuario con los permisos adecuados será capaz de listar los accesos realizados en la entidad.**	
Observaciones: Los accesos no podrán ser eliminados o modificados.	

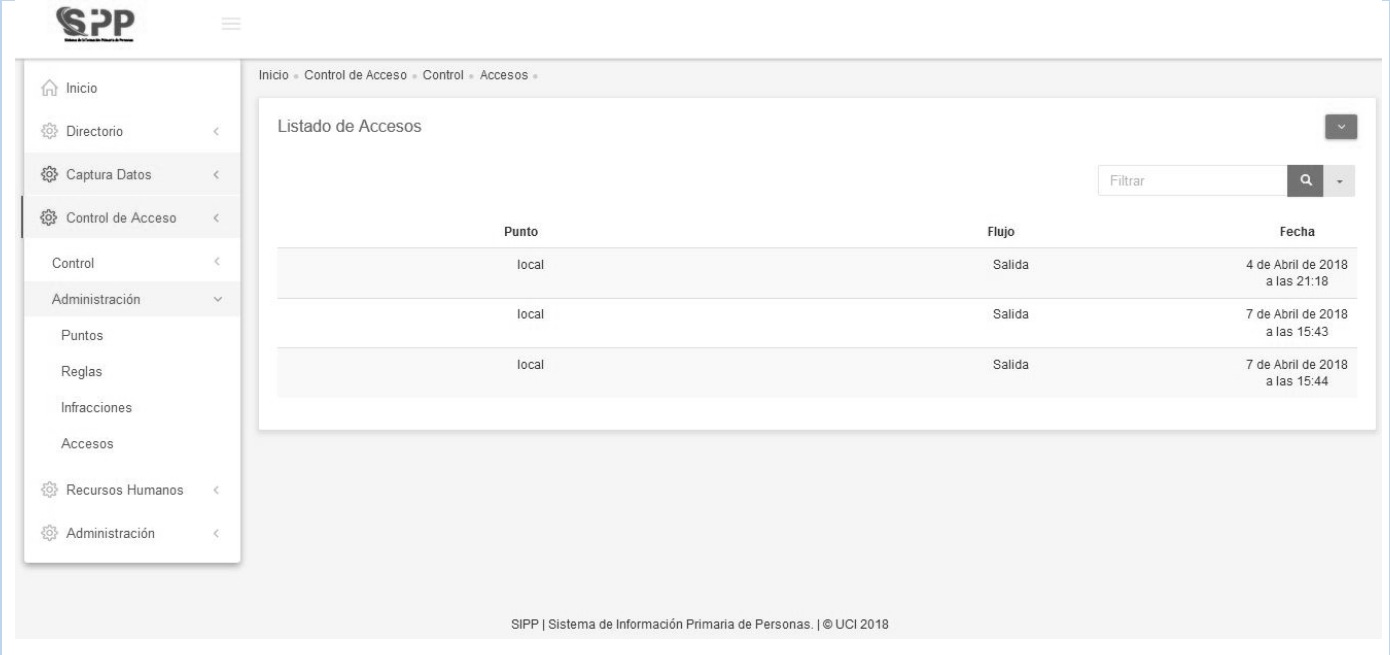


Tabla 17: HU-7 Realizar reportes de Accesos.

Número: HU-7	**Nombre de requisito: Realizar reportes de Accesos**
Programador: Christian M. Paneque	**Iteración Asignada:** 1
Prioridad: Baja	**Tiempo:** 32
Riesgo: Baja	**Tiempo real:** 24
Descripción: **El usuario con los permisos adecuados será capaz realizar reportes sobre los accesos de la entidad.**	
Observaciones: Los reportes serán exportados a Excel.	

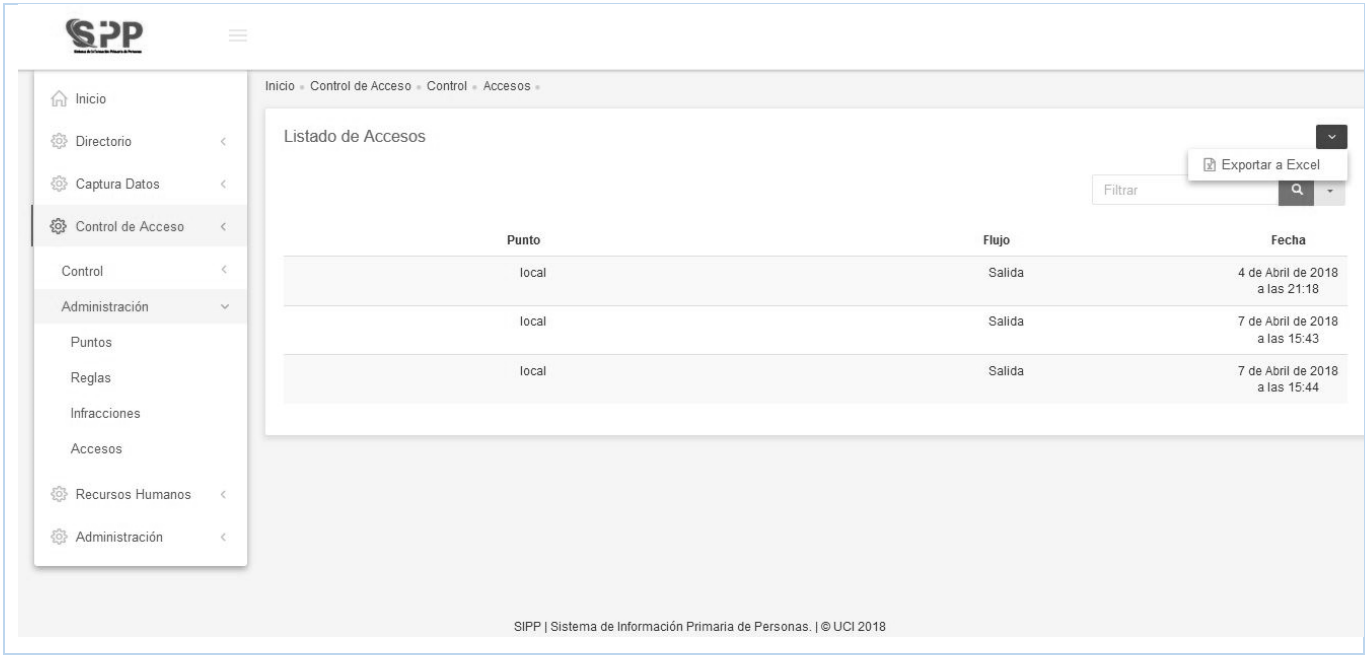
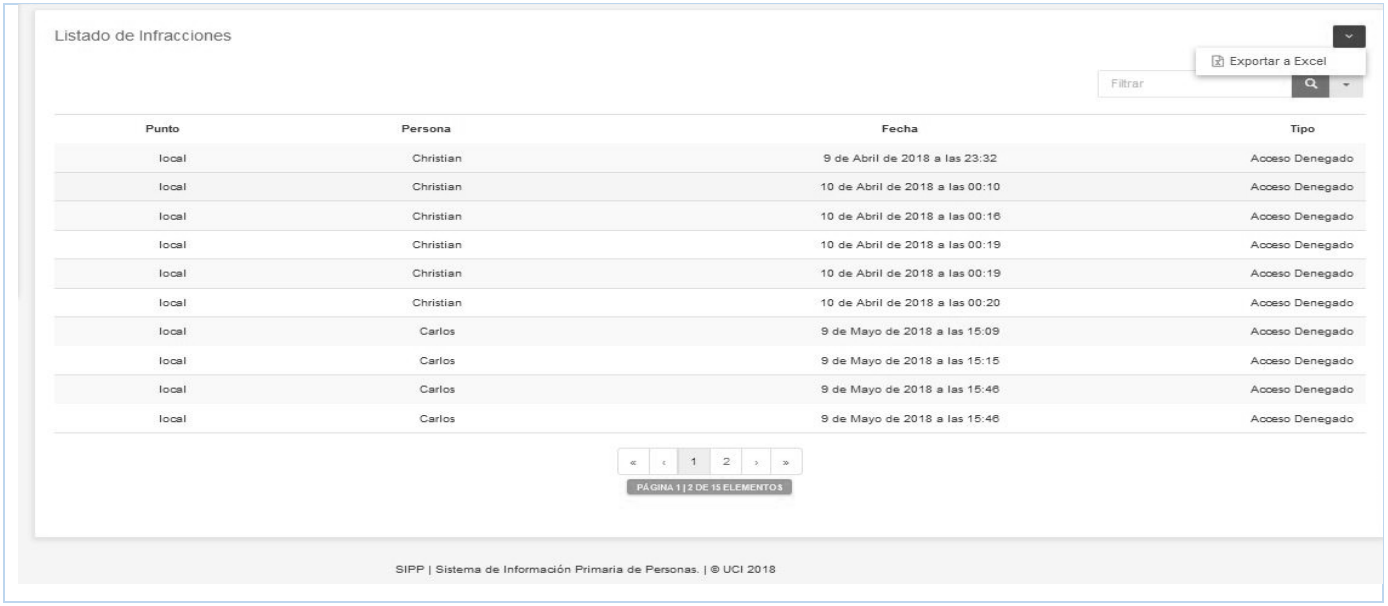


Tabla 18: HU-8 Realizar reportes de Infracciones.

Número: HU-8	**Nombre de requisito: Realizar Reportes de Infracciones**
Programador: Christian M.Paneque	**Iteración Asignada:** 1
Prioridad: Baja	**Tiempo:** 32
Riesgo: Baja	**Tiempo real:** 24
Descripción: **El usuario con los permisos adecuados será capaz realizar reportes sobre las infracciones cometidas en el acceso a las áreas de la entidad.**	
Observaciones: Los reportes serán exportados a Excel.	


Listado de Infracciones
Exportar a Excel
Filtrar
Punto
Persona
Fecha
Tipo
local
Christian
9 de Abril de 2018 a las 23:32
Acceso Denegado
local
Christian
10 de Abril de 2018 a las 00:10
Acceso Denegado
local
Christian
10 de Abril de 2018 a las 00:16
Acceso Denegado
local
Christian
10 de Abril de 2018 a las 00:19
Acceso Denegado
local
Christian
10 de Abril de 2018 a las 00:19
Acceso Denegado
local
Christian
10 de Abril de 2018 a las 00:20
Acceso Denegado
local
Carlos
9 de Mayo de 2018 a las 15:09
Acceso Denegado
local
Carlos
9 de Mayo de 2018 a las 15:15
Acceso Denegado
local
Carlos
9 de Mayo de 2018 a las 15:46
Acceso Denegado
local
Carlos
9 de Mayo de 2018 a las 15:46
Acceso Denegado
« ‹ 1 2 › »
PÁGINA 1 | 2 DE 15 ELEMENTOS
SIPP | Sistema de Información Primaria de Personas. | © UCI 2018

Anexo 3- Pruebas Funcionales.

Tabla 19: Variables para los casos de prueba funcional Gestionar Puntos (Elaboración propia)

No.	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Nombre	Campo de texto	No	Se introduce el nombre del punto.	NP
2	Fecha	Automático	No	Se obtiene del sistema.	F
3	IP	Campo de IP	No	Se introduce una dirección IP.	IP
4	Área	Campo de selección múltiple	No	Se selecciona el o las áreas a controlar.	A
5	Regla	Campo de selección	No	Se selecciona la regla de comportamiento del punto.	R

Tabla 20: Caso de prueba funcional Insertar Punto ESC 1 (Elaboración propia)

Descripción general								
Permitirá insertar un nuevo punto								
Condiciones de ejecución								
Para insertar un nuevo punto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	**NP**	**F**	**IP**	**A**	**R**	**R/ del Sistema**	**Flujo central**
ESC 1.1 Introducir datos del punto correctamente**.**	El usuario llena y activa todos los campos del formulario.	Nuevo	Automático	127.0.0.1	CISED	Nueva	El sistema crea un punto satisfactoriamente y muestra notificación de éxito.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso - Administración Punto-

								Nuevo". Aparecerá un formulario con los campos a llenar.

Tabla 21: Caso de prueba funcional Modificar Punto ESC 1 (Elaboración propia)

Descripción general								
Permitirá modificar un punto.								
Condiciones de ejecución								
Para modificar un nuevo punto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	**NP**	**F**	**IP**	**A**	**R**	**R/ del sistema**	**Flujo central**
ESC 1.2 Modificar datos del punto correctamente**.**	El usuario selecciona el botón modificar.	Nuevo1	Automático	128.0.0.1	UCI	Nueva	El sistema modifica el punto satisfactoriamente y muestra notificación de éxito.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso - Administración Punto-Modificar.

Tabla 22: Caso de prueba funcional Eliminar Punto ESC 1 (Elaboración propia)

Descripción general								
Permitirá eliminar un punto.								
Condiciones de ejecución								
Para eliminar un punto el usuario debe estar logueado en el sistema y poseer los permisos necesarios. Debe existir algún punto creado.								
Escenario	**Descripción**	**NP**	**F**	**IP**	**A**	**R**	**R/ del Sistema**	**Flujo central**
ESC 1.3 Eliminar el punto correctamente**.**	El usuario selecciona el botón eliminar.	N/S	N/S	N/S	N/S	N/S	El sistema elimina el punto satisfactoriamente y muestra notificación de éxito.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso - Administración Puntos-Eliminar.

Tabla 23: Caso de prueba funcional Insertar Punto ESC 2 (Elaboración propia)

Descripción general								
Permitirá insertar un nuevo punto.								
Condiciones de ejecución								
Para insertar un nuevo punto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	**NP**	**F**	**IP**	**A**	**R**	**R/ del sistema**	**Flujo central**
ESC 2.1 Introducir datos del punto vacíos**.**	El usuario no llena los campos del formulario.	vacío	Automática	vacío	vacío	vacío	El sistema cancela la inserción del punto y muestra un mensaje de error “Este campo es necesario”, para los campos nombre, IP, área y regla.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Puntos- Nuevo". Aparecerá un formulario con los campos a llenar.

Tabla 24: Caso de prueba funcional Modificar Punto ESC 2 (Elaboración propia)

Descripción general								
Permitirá modificar un punto.								
Condiciones de ejecución								
Para modificar un punto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	**NP**	**F**	**IP**	**A**	**R**	**R/ del sistema**	**Flujo central**
ESC 2.2 Introducir datos del punto vacíos**.**	El usuario selecciona el botón modificar e intenta introducir los datos del punto vacíos.	vacío	Automático	vacío	vacío	vacío	El sistema cancela la inserción del punto y muestra un mensaje de error “Este campo es necesario”, para los campos nombre, IP, área y regla.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Puntos- Modificar.

Tabla 25: Caso de prueba funcional Insertar Punto ESC 3 (Elaboración propia)

Descripción general								
Permitirá insertar un nuevo punto.								
Condiciones de ejecución								
Para insertar un nuevo punto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	**NP**	**F**	**IP**	**A**	**R**	**R/ del sistema**	**Flujo central**
ESC 3.1 Introducir datos del punto no válidos**.**	El usuario introduce caracteres numéricos en el campo Nombre	34141	Auto	129.0.0.1	CISED	Nueva	El sistema cancela la inserción del punto y muestra un mensaje de error “Caracteres no válidos”, para el campo nombre.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Punto-Nuevo". Aparecerá un formulario con los campos a llenar.

Tabla 26: Caso de prueba funcional Modificar Punto ESC 3 (Elaboración propia)

Descripción general								
Permitirá modificar un Punto.								
Condiciones de ejecución								
Para insertar una nueva regla el usuario debe estar logueado en el sistema y poseer los permisos necesarios.								
Escenario	**Descripción**	NP	F	**IP**	**A**	**R**	**R/ del sistema**	**Flujo central**
ESC 3.2 Introducir datos del Punto no válidos**.**	El usuario introduce caracteres numéricos en el campo Nombre	34141	Automático	192.0.0.1	UCI	Nueva	El sistema cancela la inserción del Punto y muestra un mensaje de error “Caracteres no válidos”, para el campo nombre.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Administración Puntos-Modificar". Aparecerá un formulario con los campos a llenar.

Tabla 27: Variables para los casos de prueba funcional Controlar Acceso (Elaboración propia)

No.	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	CI	Campo de texto	Si	Se introduce un CI.	CI
2	Solapín	Campo de texto	Si	Se introduce un solapín.	S

Tabla 28: Caso de prueba funcional Controlar Acceso ESC 1 (Elaboración propia)

Descripción general					
Permitirá controlar el acceso.					
Condiciones de ejecución					
Para controlar el acceso el usuario deberá tener permisos de control .					
Escenario	**Descripción**	**CI**	**S**	**R/ del Sistema**	**Flujo central**
ESC 1 Introducir datos del formulario válidos**.**	El usuario introduce un número de solapín o CI.	79	Vacio	El sistema realiza una búsqueda del CI en el sistema, si el CI existe muestra los datos, si no muestra el mensaje “No existe”, si la persona con ese CI tiene acceso al área muestra el mensaje “Acceso Permitido”, de lo contrario muestra “Acceso Denegado”	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso -Control Acceso ". Aparecerá un formulario con los campos a llenar.

Tabla 29: Caso de prueba funcional Controlar Acceso ESC 2 (Elaboración propia)

Descripción general					
Permitirá controlar el acceso.					
Condiciones de ejecución					
Para controlar el acceso el usuario deberá tener permisos de control .					
Escenario	**Descripción**	**CI**	**S**	**R/ del Sistema**	**Flujo central**
ESC 2 Introducir datos del formulario vacíos**.**	El usuario no introduce un número de solapín o CI.	Vacío	Vacío	El sistema muestra el mensaje “Debe llenar al menos un campo”	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso -Control Acceso ". Aparecerá un formulario con los campos a llenar.

Tabla 30: Caso de prueba funcional Controlar Acceso ESC 3 (Elaboración propia)

Descripción general

Permitirá controlar el acceso.					
Condiciones de ejecución					
Para controlar el acceso el usuario deberá tener permisos de control .					
Escenario	**Descripción**	**CI**	**S**	**R/ del Sistema**	**Flujo central**
ESC 3 Introducir datos del formulario no válidos**.**	El usuario introduce letras en los campos solapín o CI	wsxcfv	Vacío	El sistema muestra el mensaje “Caracteres no válidos”	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso -l Acceso ". Aparecerá un formulario con los campos a llenar.

Tabla 31: Variables para los casos de prueba funcional Insertar Visitante (Elaboración propia)

No.	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Nombre	Campo de texto	No	Se introduce el nombre del visitante.	NV
2	Apellidos	Campo de texto	No	Se introducen los apellidos del visitante.	AV
3	CI	Campo de texto	No	Se introduce el CI del visitante.	CI
4	Sexo	Campo de selección	No	Se selecciona el sexo del visitante.	S
5	Municipio	Campo de selección	No	Se selecciona el sexo del visitante.	M
	Provincia	Campo de selección	No	Se selecciona la provincia del visitante.	P
	Solapín	Campo de selección	No	Se selecciona el solapín que se le asignará al visitante.	SO

Tabla 32: Caso de prueba funcional Insertar Visitante ESC 1 (Elaboración propia)

Descripción general
Permitirá insertar un nuevo visitante
Condiciones de ejecución
Para insertar un nuevo Visitante el usuario debe estar logueado en el sistema y poseer los permisos necesarios.

Escenario	Descripción	NV	AV	CI	S	M	P	SO	R/ del Sistema	Flujo central
ESC 1 Introducir datos del visitante correctamente**.**	El usuario llena y activa todos los campos del formulario.	Nuevo	Visitante	55	Masculino	Matanzas	Matanzas	**55**	El sistema crea un visitante satisfactoriamente y muestra notificación de éxito.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso - Control – Visitantes- Nuevo". Aparecerá un formulario con los campos a llenar.

Tabla 33: Caso de prueba funcional Insertar Visitante ESC 2 (Elaboración propia)

Descripción general										
Permitirá insertar un nuevo visitante										
Condiciones de ejecución										
Para insertar un nuevo Visitante el usuario debe estar logueado en el sistema y poseer los permisos necesarios.										
Escenario	**Descripción**	**NV**	**AV**	**CI**	**S**	**M**	**P**	**SO**	**R/ del Sistema**	**Flujo central**
ESC 2 Introducir datos del visitante vacíos**.**	El usuario llena y activa todos los campos del formulario vacíos.	vacío	vacío	vacío	vacío	vacío	vacío	vacío	El sistema cancela la inserción del visitante y muestra un mensaje de error "Este campo es necesario", para los campos.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso - Control – Visitantes- Nuevo". Aparecerá un formulario con los

										campos a llenar.

Tabla 34: Caso de prueba funcional Insertar Visitante ESC 3 (Elaboración propia)

Descripción general										
Permitirá insertar un nuevo visitante										
Condiciones de ejecución										
Para insertar un nuevo Visitante el usuario debe estar logueado en el sistema y poseer los permisos necesarios.										
Escenario	**Descripción**	**NV**	**AV**	**CI**	**S**	**M**	**P**	**SO**	**R/ del Sistema**	**Flujo central**
ESC 3 Introducir datos del visitante con caracteres extraños**.**	El usuario llena y activa todos los campos del formulario vacíos.	13	432	542	N/P	N/P	N/P	N/P	El sistema cancela la inserción del visitante y muestra un mensaje de error “Caracteres no válidos”, los campos.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso - Control – Visitantes- Nuevo". Aparecerá un formulario con los campos a llenar.

Tabla 35: Caso de prueba funcional Listar Infracciones (Elaboración propia)

Descripción general			
Permitirá listar las Infracciones.			
Condiciones de ejecución			
Para listar las infracciones el usuario deberá tener permisos de administración.			
Escenario	**Descripción**	**R/ del Sistema**	**Flujo central**
Listar Infracciones	El usuario intenta listar las infracciones	El sistema lista correctamente las infracciones.	En el menú a la izquierda se sigue la siguiente ruta: “Control de Acceso -l Administración- Infracciones ".

Tabla 36: Caso de prueba funcional Listar Accesos (Elaboración propia)

Descripción general			
Permitirá listar los accesos.			
Condiciones de ejecución			
Para listar los Accesos el usuario deberá tener permisos de administración.			
Escenario	**Descripción**	**R/ del Sistema**	**Flujo central**
Listar accesos	El usuario intenta listar los Accesos	El sistema lista correctamente los accesos.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso -l Administración-Accesos ".

Tabla 37: Caso de prueba funcional Realizar reportes de Infracciones (Elaboración propia)

Descripción general			
Permitirá realizar reportes de infracciones.			
Condiciones de ejecución			
Para realizar reportes de infracciones el usuario deberá tener permisos de administración.			
Escenario	**Descripción**	**R/ del Sistema**	**Flujo central**
Realizar reportes de Infracciones	El usuario exporta a Excel un reporte de las infracciones.	El sistema exporta a Excel la información de las infracciones.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso -l Administración-Infracciones-Exportar a Excel ".

Tabla 38: Caso de prueba funcional Realizar reportes de Accesos (Elaboración propia)

Descripción general			
Permitirá realizar reportes de accesos.			
Condiciones de ejecución			
Para realizar reportes de accesos el usuario deberá tener permisos de administración.			
Escenario	**Descripción**	**R/ del Sistema**	**Flujo central**
Realizar reportes de accesos	El usuario exporta a Excel un reporte de Accesos.	El sistema exporta a Excel la información de los accesos.	En el menú a la izquierda se sigue la siguiente ruta: "Control de Acceso -l Administración-Accesos-Exportar a Excel ".

Anexo 4-Valoración de Expertos.

Tabla 39: Valoración de expertos.

Valoración Sentencia No.	Osay Gonzáles Fuentes	José Carlos Pérez Zamora	Miguel Jaeger Rodríguez Lazo	Ernesto Soto Gómez	Denier Naranjo Oliva
1	Muy adecuado	Muy adecuado	Muy adecuado	Muy adecuado	Muy adecuado
2	Muy adecuado	Muy adecuado	Muy adecuado	Muy adecuado	Muy adecuado
3	Muy adecuado	Muy adecuado	Muy adecuado	Adecuado	Adecuado
4	Muy adecuado	Muy adecuado	Muy adecuado	Muy adecuado	Muy adecuado
5	Muy adecuado	Muy adecuado	Muy adecuado	Muy adecuado	Adecuado