UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 1

Sistema de control de acceso para entidades

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

Autor: Cléusio de Oliveira Carlos

Tutores: Ing Dairo Roberto Gil Martín

Ing. Henrry Díaz Torres

La Habana, junio de 2020

Declaración de autoría

Declaro por este medio que yo Cleusio De Oliveira Carlos, con carné de identidad 94061001504 soy el autor principal del trabajo titulado Sistema de control de acceso para entidades y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año 2020.

Autor

_____

Cléusio de Oliveira Carlos

Tutor

_____

Ing. Dairo Roberto Gil Martín

Tutor

_____

Ing. Henrry Díaz Torres

Agradecimiento

Agradezco principalmente a Dios por darme la sabiduría, la fuerza, el corazón y las ganas, agradezco a mi madre que me apoya en el camino de la vida y que siempre vela por mi bienestar, a mi familia por su apoyo incondicional, tanto en los buenos como en los malos momentos, motivación y cariño.

A todos y cada uno de mis profesores de la UCI que sin duda alguna me han hecho crecer intelectualmente. Y en especial a profesora Yanelis Benítez, por su atención y cariño que siempre ha tenido conmigo y por su ayuda y consejos.

Resumen

La investigación está encaminada a obtener una herramienta de control de acceso para entidades con el objetivo de tener el control del acceso físico y lógico a estaciones de trabajo. Actualmente el hombre ha encontrado en la tecnología un instrumento capaz de ayudarle a satisfacer esta necesidad de manera segura y a bajo precio a través de un sistema. Este sistema de control de acceso podrá identificar las personas mediante el ingreso de su código de barras o su número de credencial, permitir el acceso o denegarlo y registrar las entradas y salida de las personas, así como detectar posibles violaciones de su seguridad y/o la posible entrada de personas ajenas a la institución sin autorización, y al mismo tiempo debe ser capaz de generar una credencial para cualquier tipo de entidad. Deberá gestionar la entidad y ajustar las personas circuladas, las estadísticas de accesos, los horarios, las credenciales de los puntos de acceso a una entidad. Dicho Sistema aplica seguridad de negocio a las fuentes datos de personas, es decir, que a estas personas solo se le mostrará la información de la entidad a la que pertenece, por cada institución y/o entidad solo se pondrá ver las personas, puntos de accesos, personas circuladas, horarios y credenciales asociadas a dicha entidad.

Palabras Claves: control de acceso, acceso autorizado, autenticación, seguridad.

Índice

Introducción

En la actualidad, el aumento del desarrollo tecnológico ha propiciado con el desarrollo de la sociedad de la información, una nueva revolución social, con el desarrollo de la sociedad de la información. Con ello, se desea hacer referencia a que la materia prima que es la información será el motor de esta nueva sociedad, y en torno a ella, surgirán profesiones y trabajos nuevos, o se readaptarán las profesiones existentes.

La dimensión social de las tecnologías de la información y la comunicación (TIC) se vislumbra atendiendo a la fuerza e influencia que tiene en los diferentes ámbitos y a las nuevas estructuras sociales que están emergiendo, produciéndose una interacción constante y bidireccional entre la tecnología y la sociedad. La influencia de la tecnología sobre la sociedad ha sido claramente explicitada por Melvin Kranzberg, en su ley sobre la relación entre tecnología y sociedad: La tecnología no es buena ni mala, ni tampoco neutral, pero esta relación no debe entenderse como una relación fatalista y determinista, sino que a nuestro entender nos conduce a nuevas situaciones y planteamientos que deben llevarnos a través de la investigación y el análisis de sus efectos a tomar posiciones que marquen el camino y la dirección a seguir atendiendo a la sociedad que deseamos construir. (Technology and Culture, 1986).

Conforme crece la utilización de las TIC crece la necesidad de ofrecer a las organizaciones mecanismos para una mejor implementación de la seguridad de la información. Dichos mecanismos deben permitir controlar el acceso a los recursos, gestionar usuarios y sus datos de identificación, asociar roles, perfiles y políticas de seguridad. Estos suelen estar formados por dispositivos de autenticación que facilitan el control del acceso lógico de los usuarios en los sistemas informáticos debido a que la ausencia de seguridad pone en riesgo la información y conlleva a considerables pérdidas monetarias.

En busca de una solución factible para garantizar que la gestión de la seguridad de la información se realice mediante un proceso sistemático, documentado y conocido por toda la organización, se define como política estándar la ISO 27001[1]. Uno de los conceptos fundamentales asociados con la seguridad de la información precisamente está asociado al control de acceso.

Un control de accesos es un sistema electrónico que restringe o permite el acceso de un usuario o grupo de usuarios a un área específica validando la identificación por medio de diferentes tipos de lectura (clave por teclado, lector de tarjetas, biometría, etc.) y a su vez controlando el recurso (puerta, armario, etc.) por medio de un dispositivo eléctrico como un electroimán, pestillo o motor. Un control de accesos requiere flexibilidad para que no haya limitaciones en la movilidad por cambios que se producen en los permisos. Necesita precisión para que se le asigne el permiso correcto a cada persona. Y también es necesario que tenga suficiente capacidad para almacenamiento y registro de un mínimo de datos (PEREZ, 2016).

Hasta no hace mucho tiempo el acceso a zonas restringidas se realizaba mediante medios mecánicos (Cerrojos, llaves, etc.), pero hoy en día están siendo sustituidos por sistemas más seguros y modernos basados en dispositivos electrónicos.

La función principal del control de accesos es la de controlar entradas y salidas libremente de las personas a diversas áreas que se denominan protegidas. Este sistema brinda información acerca de quién entra, cuando entra y a dónde entra cada individuo. Se pueden encontrar modelos estándar (sistema de control en el pomo exterior de la puerta) o modelos duales, con lectura por ambos lados para zonas donde se necesita acceso con distintivo a ambos lados de la puerta, como zonas de paso o pasillos (PEREZ, 2016).

[1] El Sistema de Gestión de Seguridad de la Información (SGSI) es el concepto central sobre el que se construye ISO 27001. Garantiza que los riesgos de la seguridad de la información sean conocidos, asumidos, gestionados y minimizados por la organización de una forma documentada, sistemática, estructurada, repetible, eficiente y adaptada a los cambios que se produzcan en los riesgos, el entorno y las tecnologías.

.

La necesidad del control de acceso está condicionada por la obligación de controlar el acceso físico no autorizado a una entidad, resulta complicado mantener el control de entrada y salida de personal autorizado en los locales de la entidad debido a la gran masa de trabajadores y visitantes, lo que figura un riesgo de seguridad para la entidad en cuanto a información y tecnología existentes en la misma. En algunos casos se minimiza la probabilidad de ocurrencia de un acceso no autorizado a través de medidas técnicas. Para otros casos es posible garantizarlo cuando se realiza el control de forma presencial, evitando así que los usuarios no autorizados hagan uso de la estación de trabajo, el control de entrada y salida completamente automatizados con diferentes tipos de tecnologías y dispositivos. Es importante realizar un estudio adecuado, segmentando las zonas, los grupos de acceso, los horarios permitidos, el nivel de acceso de cada usuario, medir la cantidad de personas o vehículos que transitan por cada zona y establecer claramente los objetivos de cada control de acceso.

Es por ello que se propone un sistema de control de acceso capaz de identificar a las personas mediante el ingreso de su código de barras o su número de solapin, permitiendo o denegando el acceso a la entidad, además de registrar el horario de entrada y salida de las personas, así como detectar posibles violaciones de su seguridad y/o la posible entrada de personas ajenas a la entidad sin autorización. Dicho sistema debe generar una credencial según la entidad a la que se desea acceder; entidad que deberá especificar las personas circuladas, las estadísticas de accesos y los horarios. Además, debe aplicar seguridad de negocio a las fuentes datos de personas “es decir” que a estas personas solo se le mostrará la información de la entidad a la que pertenece, que a su vez dicha entidad solo podrá ver las personas, puntos de accesos, personas circuladas, horarios y credenciales asociadas a ella.

Luego de analizar la problemática anterior, se plantea la siguiente interrogante como **Problema de investigación:** ¿Cómo contribuir a la gestión del control de acceso en las entidades?

Para dar solución al problema antes expuesto se formula el siguiente **objetivo general:** Desarrollar un sistema para el control de acceso en entidades.

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

- Fundamentar la selección de la metodología, herramientas y tecnologías a utilizar en el desarrollo del sistema para el control de acceso en entidades.
- Realizar un análisis detallado sobre los modelos de control de acceso existentes para poder seleccionar el más adecuado.
- Seleccionar la base tecnológica necesaria para la implementación del sistema de control de acceso en entidades.
- Realizar el análisis y diseño del sistema para el control de acceso en entidades.
- Implementar el sistema para el control de acceso en entidades.
- Realizar pruebas al sistema para el control de acceso en entidades.

Teniendo como **objeto de estudio:** El control de acceso físico a estaciones de trabajo.

Enmarcado en el **campo de acción:** Control de acceso físico a estaciones de trabajo del centro de tecnologías para la formación.

Con el propósito de cumplir con el objetivo general, se plantean las siguientes **tareas de la investigación:**

- Resumir sobre la fundamentación asociada al concepto de control de acceso físico.
- Análisis comparativo de los sistemas informáticos para el control de acceso físico.
- Realización de un estudio sobre el uso de la tecnología en función del desarrollo de la habilidad técnica diseñar.
- Identificación de los requerimientos asociados a una posible solución, a partir del estudio de herramientas similares.
- Caracterización de la habilidad diseñar bases de datos partiendo de las teorías de diseño de Bases de Datos Relacionales.
- Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
- Implementación de la solución propuesta para el control de acceso físico a la entidad.

- Selección de los tipos de pruebas a realizar sobre el sistema.
- Realización de las pruebas seleccionadas sobre el sistema.

Métodos científicos de investigación.

Entre los **métodos teóricos** usados para esta investigación está el **histórico-lógico** para el estudio del avance de los controladores de acceso automatizados, con el fin de apoyar y facilitar el proceso de modelado.

El **hipotético-deductivo** para poder apropiarse adecuadamente del negocio y realizar procedimientos que ayuden a su comprensión y a la deducción de situaciones que complejizan el problema, elementos vitales para la obtención de una solución efectiva. El **método empírico** cuantitativo usado fue la **observación** con el fin de recopilar datos a medida que se desarrolla el software.

El presente trabajo de diploma presenta la siguiente estructura**:**

Capítulo 1

En este capítulo se abordan aspectos generales del estudio del estado del arte, se explicará cómo se ejecuta el proceso de control de acceso en una entidad. Se apuntará los principales problemas que se están enfrentando ahora en las entidades y con base a estos problemas se buscará una solución óptima para su cumplimiento. Se hace una descripción y selección de las herramientas, tecnologías y la metodología a utilizar en el desarrollo de la aplicación.

Capítulo 2

Se describen de manera general los procesos involucrados en el campo de acción, se presenta una descripción general de la propuesta de solución y su funcionamiento, se muestra la solución que se propone, se especifican los requisitos funcionales, y no funcionales que se requiere, los actores del sistema, diagramas de caso de uso del sistema y el patrón de casos de uso utilizado.

Capítulo 3

Se representan los elementos físicos necesarios para el despliegue de la aplicación, empleando para ello un diagrama de despliegue. Se presentan los aspectos fundamentales de la fase de construcción, dentro de la que destacan los procesos de implementación y prueba de software, se definen los tipos de pruebas que se le realizarán al software y los resultados de las mismas.

Capítulo 1: Fundamentación teórica

Introducción

En el presente capítulo se exponen los fundamentos teóricos que sustentan la base de la investigación para el desarrollo del sistema de control de acceso para entidades. Además, se realiza una breve descripción de las distintas herramientas, lenguajes y tecnologías que serán utilizadas para el desarrollo del mismo especificándose también la metodología de desarrollo de software a utilizar.

1.1. Conceptos fundamentales

Definición control de acceso proceso de conceder permisos a usuarios o grupos de acceder a objetos tales como ficheros o impresoras en la red. El control de acceso está basado en tres conceptos fundamentales: identificación, autenticación y autorización.

Control acceso se trata sobre los sistemas que protegen a los objetos de valor y también sobre las decisiones tomadas por las personas que determinan quién recibe alguna clase de acceso. El control de acceso puede ser utilizado para controlar el acceso a espacios físicos o a la información dentro de un sistema, (Peltier, 2014)[2].

Define que los sistemas de control de acceso se ponen en marcha para garantizar que sólo las personas autorizadas tengan acceso a la información, y así garantizar que la misma se mantenga intacta y disponible cuando sea necesario. El propósito de los sistemas de control de acceso es evitar la modificación de la información por los usuarios no autorizados, permitir la modificación de la información por los usuarios autorizados, y preservar la consistencia interna y externa de los datos.

[2] Thomas R. Peltier 2014. Information Security Fundamentals, 2a ed. EE. UU

Los sistemas de control en función de su grado de automatización se clasifican en:

Controles manuales: estos sistemas se basan en que son los vigilantes, guardias de seguridad, personal administrativo y/o recepcionistas quienes dan o deniegan el permiso de acceso. Para que este sistema funcione, se requiere un gran esfuerzo y planificación de las personas encargadas, es por ello que lo adecuado sería que el personal que está a cargo de él, conociese a todas las personas autorizadas para acceder al lugar. El problema de este tipo es que no funciona cuando el grupo autorizado es muy grande o cuando el personal cambia a menudo.

Controles semimanuales: este tipo utiliza equipos o elementos electromecánicos para apoyar al personal en la evaluación de la solicitud de acceso y en la toma de decisión para permitir o denegar la entrada. Los elementos o dispositivos más utilizados son las botoneras digitales.

Controles automáticos: son aquellos en los cuales las etapas de verificación y acceso son efectuadas enteramente por equipos o sistemas electrónicos, los cuales están programados para tomar decisiones cuando alguien lo requiere. (PEREZ, 2016).

Para lograr esto, se aplican los controles. Los controles ayudan a mitigar el riesgo y reducir la posibilidad de pérdida, y requiere de las combinaciones de controles para una defensa en profundidad. Una forma de clasificar a los controles de acceso es mediante la descripción en la forma en que se implementan. Los tres tipos diferentes de implementación son: administrativos, físicos y técnico / lógicos (Peltier, 2014).

1.1.1. Tipos de Controles

Controles administrativos

Los controles administrativos ayudan a hacer frente a las amenazas internas, como el robo de información privilegiada o violación a bases de datos. Estos controles pueden ser las políticas, procedimientos, capacitaciones, revisiones, etc., que establece la organización, (Peltier, 2014).

Controles físicos

Los controles físicos se utilizan para disuadir y prevenir eventos desastrosos dentro de un ambiente físico, puede ser tales como guardias de seguridad, cámaras de seguridad, asegurando de salas de servidores, el bloqueo de los ordenadores portátiles (Peltier, 2014).

Controles técnicos o lógicos

Los controles técnicos o lógicos restringen el acceso a los sistemas de información y protegen la información que ellos contienen; tales como el cifrado, tarjetas inteligentes, listas de control de acceso, protocolos de transmisión (Peltier, 2014).

1.1.2. Conceptos asociados al dominio del problema

Para desarrollar aplicaciones informáticas fiables, que ofrezcan soluciones integrales a los problemas planteados por los clientes o la ciencia, se hace indispensable que los equipos de desarrollo conozcan a fondo el negocio. El dominio de los principales conceptos, flujos de información y filosofía de trabajo hacen posible la obtención de soluciones robustas.

Entre estos conceptos que son necesarios conocer de acuerdo con el dominio de estudio en este caso particular, es necesario abordar sobre el término “seguridad informática”, pues tiene una relación directa con el control de acceso, siendo este uno de los conceptos fundamentales a implementar en un SCA, por esta razón a continuación de forma conceptual se abordarán conceptos relacionados con el término en cuestión.

1.1.3. Seguridad informática

La seguridad de la información protege a la información de un amplio rango de amenaza para asegurar la continuidad del negocio, minimizar los daños a la organización y maximizar el retorno de las inversiones y oportunidades del negocio, preservando la confidencialidad, integridad y disponibilidad de la información, así mismo, otras propiedades como la autenticidad, no rechazo, contabilidad y confiabilidad también pueden ser consideradas.

Según el Dr. Jorge Ramió Aguirre la seguridad informática es el estado de cualquier tipo de información (sea informática o no) que indica que el sistema esté libre de peligro, daño o riesgo. Se concibe como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo, (Seguridad informática, 2010).

1.1.4. Características de la seguridad informática

- **Integridad:** los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.

- **Confidencialidad** la ISO define la confidencialidad como la propiedad para garantizar que la información se pueda acceder únicamente por aquellas entidades autorizadas para hacerlo.

- **Disponibilidad:** los activos informáticos son accedidos por las personas autorizadas en el momento requerido.

- **Irrefutable:** el uso y/o modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción.

Por otra parte, el Dr. Charles P. Pfleeger plantea que es el conjunto de métodos y herramientas destinados a proteger los bienes (o activos) informáticos de una institución. La información es el activo más preciado y la seguridad en la información tiene el objetivo de garantizar: confidencialidad, integridad y disponibilidad. A partir de estas definiciones se puede concluir que la seguridad informática es el estado de cualquier tipo de información que indica que está libre de peligro, daño o riesgo. Implica la preservación de la: integridad, confidencialidad, disponibilidad, así como la condición de ser irrefutable y se logra implementando un conjunto adecuado de controles que abarcan políticas, prácticas, procedimientos, estructuras organizacionales y funciones de software. (pfleeger, et al., 2010).

Por lo antes planteado se puede decir que es esencial tener en cuenta los pilares fundamentales de la seguridad Informática conceptualmente expuestos, tanto en la problemática que se pretende solucionar como en el desarrollo de un software, especialmente en un sistema para el control de acceso.

1.2. Modelo de control de acceso

Según el tipo de política de autorización, los modelos de control de acceso pueden ser divididos en: modelos de control de acceso tradicional, modelo de control de acceso basado en roles y modelo de control de acceso basado en atributos.

Modelos tradicionales de control de acceso

Existen dos tipos de modelos tradicionales de control de acceso, los cuales son: el modelo de control de acceso discrecional o Discretionary Access Control (**DAC**), y el modelo de control de acceso mandatario o Mandatatoryl Access Control (**MAC**).

Modelo de control de acceso discrecional(DAC)

Yang-Feng describe que la idea principal del modelo DAC, es que el sujeto (se 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, et al., 2013).

Modelo de control de acceso mandatario(MAC)

Por otro lado, Tounis describe que 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 hacia alguna información de los objetos. Con el fin de garantizar el acceso a los recursos 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. (YA, 2014).

Modelo de control de acceso basado en roles(RBAC)

Yang-Feng describe que el control de acceso basado en roles o Role Based Access Control, simplifica la gestión de autorización en diferentes ambientes. En los sistemas MAC o DAC, los permisos de acceso se conceden directamente al usuario. Mientras que el número de usuarios en el sistema es grande y cambiante, la complejidad en la gestión de autorización aumenta. El control de acceso basado en roles asigna los derechos de acceso a un rol. Los roles son relativamente estables en comparación a los usuarios. El rol de hecho es asociado con un conjunto de opciones de permisos en particular. Cuando los usuarios cambian, los roles solo necesitan ser retirados y reasignados. (J, et al., 2013).

Del mismo modo, Tounis en (2014) describe que el modelo de control de acceso basado en roles, es considerado como una forma natural de controlar el acceso a los recursos en las organizaciones y empresas. La motivación detrás de modelo de control de acceso basado en roles parte de considerar que la responsabilidad de un sujeto es más importante que el sujeto en sí. En el modelo, un sujeto puede tener más de un rol o ser miembro de varios grupos. Finalmente, el modelo de control de acceso basado en roles tiene muchas ventajas en comparación con los otros modelos, sin embargo, tiene sus propias dificultades cuando se despliega en el mundo real. En primer lugar, la selección de los roles correctos que representan a un sistema no es una tarea fácil, y la división de los sujetos en categorías basadas en los roles podría empeorar las cosas. Los roles en el modelo, clasifican a los sujetos en una serie de categorías; así, cada sujeto tiene que tener un rol con el fin de acceder al sistema. A pesar de eso, los roles pueden dar a un sujeto más derechos que los que necesita necesariamente tener, con la posibilidad de tener otro rol que podría conducir a la violación de una política de acceso (YA, 2014).

Modelo de control de acceso no discrecional (NDAC)

Las políticas administrativas determinan quién está autorizado a codificar los controles de acceso y solo tienen sentido en políticas discrecionales. En las políticas de obligación, el control de acceso se basa completamente en la clasificación de seguridad de los sujetos y objetos, los permisos asignados no se pueden modificar una vez se han instanciado. Identifica las situaciones en las que se ha concedido la autoridad a varios usuarios, pero hay controles para la delegación y propagación de esta autoridad**.** (Peltier, 2014)**.**

1.2.1. Consideraciones para un sistema de controles de acceso

Un sistema de control de acceso debe ser planeado de acuerdo con las necesidades de seguridad del espacio al cual se va a restringir y las consideraciones prácticas del mismo. Para esto se deben considerar 7 variables básicas a la hora de crear el diseño, estas variables son las siguientes:

- Tiempo de Ingreso: Es el tiempo que le toma a una persona, que desea entrar al establecimiento, atravesar todo el sistema de seguridad; este tiempo depende del tiempo que demoran en responder los dispositivos que componen el sistema como tal.
- Aislamiento: Esta variable se refiere al lugar donde se va instalar el sistema de control de acceso, y debe garantizar que el punto donde se va a instalar el sistema es el más vulnerable del perímetro defensivo.
- Efectividad del Sistema: La medición de esta variable se realiza observando el comportamiento de 4 variables: tiempo medio entre fallas, tasas de falsas aceptaciones y falsos rechazos, y la acción en caso de falla.
- Método de Cuarentena: Esta se enfoca en el procedimiento que se realiza para detener a la persona que desea entrar o salir del perímetro protegido mientras atraviesa el sistema de control de acceso.
- Incomodidad Causada: Es importante tener en cuenta que la incomodidad causada por el sistema diseñado no disminuya o anule la capacidad operativa de los elementos protegidos.
- Tráfico: Se debe tener en cuenta el tráfico de personas que afecta al sistema, no solo un promedio de tráfico como tal, sino el tráfico que se va a tener en las horas pico.
- Costo: La idea principal de esta variable es que se debe construir un sistema de control de acceso, con la tecnología necesaria de acuerdo a lo que se quiere proteger; además, el costo del sistema debe ser acorde al valor de los objetos protegidos. (Diez, 2019).

1.2.2. Últimas tendencias en los sistemas de control de acceso

Los sistemas de control de acceso tienen como objetivo permitir o denegar la entrada de determinados individuos a una zona protegida. A día de hoy, los sistemas de control de acceso forman parte de muchas organizaciones, siendo éstas cada vez más conscientes de su importancia. Los dispositivos actuales para el control de accesos automatizado utilizan tarjetas, tecnología biométrica o la combinación de ambas. De todas las técnicas biométricas, la más extendida es la huella dactilar, pero nuevos y muy cualitativos terminales se han abierto paso en el mercado, como es el caso del reconocimiento facial. Se trata de sistemas con buenas tasas de falso rechazo, más fiables y cómodos que los existentes hace unos años.

Otras tecnologías biométricas, como el reconocimiento de iris o retina, no han acabado todavía de implantarse de forma sólida debido, principalmente, al precio de sus equipos. Por otro lado, punteras tecnologías como, por ejemplo, los terminales que fusionan el reconocimiento por huella más las venas dactilares entran pisando fuerte. Esta innovadora tecnología combina los mecanismos de protección de cada una de las biometrías, siendo extremadamente robusta al fraude. En el caso de los dispositivos donde se necesita una tarjeta, la tecnología RFID (Radio Frequency IDentification – identificación por radiofrecuencia), ha desbancado a la banda magnética y al código de barras. Se trata de un sistema de almacenamiento y recuperación de datos remoto capaz de transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio.

1.3. Estudio del estado del arte

Sistemas de control de acceso estudiados a nivel internacional

Sistema de control de acceso Arquero

El sistema de control de acceso Arquero puede controlar el acceso a las diferentes dependencias de una empresa, buscando solución de seguridad integrada e integral, plataforma unificada desde la que operar y supervisar cada uno de los subsistemas de seguridad de una instalación. Software todo en uno que reúne bajo un mismo entorno gráfico y de gestión sistemas de control de accesos y presencia, intrusión, incendio, video vigilancia, audio, control de horarios y automatización de edificios. Sistema de control de acceso Arquero es una de las soluciones más completas y potentes del mercado internacional(Ramírez, et al., 2015)**.**

Políticas de Acceso

En Arquero las políticas de acceso se definen para usuarios y rutas de seguridad, de tal forma que los derechos son de una persona o grupo para llegar a una zona asegurada, estas políticas permiten determinar qué usuarios y grupos tienen derechos sobre qué itinerarios y en qué horarios, se puede exigir, en algunos horarios o accesos críticos, la presentación de la tarjeta/biometría más un PIN para robustecer la autenticación.

Sistema de control de acceso Easy way

El sistema de control de Accesos Easy Way es un seguro método destinado a controlar el ingreso y egreso de personas a todas las áreas de la empresa (es decir, control de personal). El software de control de acceso permite configurar el hardware desde la PC, controlar desde la inclusión de planos del edificio hasta generar informes y elaborar estadísticas, es tan fácil de implementar y operar que resulta verdaderamente conveniente para una empresa que requiere controlar entradas y salidas de manera rápida pero efectiva.Easy Way utiliza la huella dactilar para validar la identidad del usuario, esta característica es única para cada individuo, de modo que resulta completamente segura, al combinar el empleo de huella con el uso de tecnología sofisticada, el proceso de identificación se agiliza y se reduce la posibilidad de fallas al momento reconocer la huella puesto que éstas se actualizan constantemente.

Este software de control de personal tiene los controles de acceso totalmente integrados y en forma modular, es habituales de distintas empresas para el control de acceso de personal y para el control de acceso a instalaciones o edificios, además, es adaptable, opcionalmente, y de sofisticados requerimientos particulares que puedan llegar a solicitarse, estableciéndose así, una relación personalizada con el cliente y su software de control de accesos.

Sistema de control de acceso con Tarjetas de Código de Barras

Los sistemas de control de acceso con tarjetas de código de barras son una forma óptica para almacenar la información reconocible por los sistemas informáticos, son de apariencia similar a las tarjetas magnéticas, pero en el lugar de la banda, llevan impreso un código de barra el cual puede incluso ser protegido con una banda protectora (código oculto) que evita la duplicación de la tarjeta por fotocopia la ventaja de esta tarjeta es que al pasarla por el lector, no existe razonamiento con un cabezal, solo hay un haz de luz que lee el código en cuestión. (Alexander Correa Espinal, 2010).

Desventajas

- Invariabilidad de la información contenida en la etiqueta de código de barras.
- Las etiquetas dañadas hacen que sea difícil para el cajero escanearlas.
- el número de 12 dígitos en la etiqueta puede estar dañado hasta el punto en que no es legible.

Sistema de control de acceso con Tarjetas de RFID (Identificación por Radio Frecuencia)

En la actualidad RFID se utiliza principalmente en el rubro de seguridad, como es el caso de los cruces fronterizos, credenciales de identidad, en el control vehicular, identificación de ganado, envío de paquetes, control de equipaje en los aeropuertos y de artículos para renta o préstamo. Tecnología de Identificación por Radiofrecuencia es un método electrónico que consiste en asignar un código de información a un producto, proceso o persona y usar esta información para identificar o acceder a información adicional al respecto. Los sistemas de identificación por radio frecuencia consisten generalmente de dos componentes:

- **El transponder o tarjeta,** pequeña etiqueta electrónica (tag) que contiene un minúsculo microprocesador y una antena de radio esta etiqueta contiene un identificador único que puede ser asociado a una persona o producto. Cuando el transponder posee la alimentación interna, se le

denomina proximidad activa y cuando el transponder no tiene esa batería interna se le denomina proximidad pasiva, mayoritariamente son utilizados los sistemas de proximidad pasiva.

- **El lector**, que obtiene el identificador del transponder la tecnología del transponder se basa en la aplicación de un transmisor/receptor encapsulado el receptor se puede activar por medio de una batería incorporada (transponder activo) o puede ser alimentado por la señal enviada por el lector (transponder pasivo) el lector genera un campo magnético cuya señal de RF es captada por el receptor del chip. Éste, a su vez activará al transmisor, el cual enviará un mensaje codificado único. Este mensaje es decodificado por el lector y procesado por la computadora. (Sistema por identificacion por radio frecuencia, 2010).

Desventajas

- La información de la etiqueta puede ser variable y reutilizable.
- Identificación simultaneas de productos.
- Problemas de confiabilidad de lecturas por lo nuevo de las tecnologías.
- No tiene operario de lectura.

Sistema de control de acceso BioStar VideoPhone

BioStar es el nombre del sistema de control de acceso con conectividad de IP y seguridad biométrica del reconocido fabricante Suprema Inc. BioStar VideoPhone es una aplicación para PC que permite al operador utilizar la PC y un dispositivo ligado como un sistema de interfono, la aplicación permite al operador visualizar quién está en la puerta y permitir el acceso, si este se ha aprobado es una aplicación única y muy útil empleada para crear un sistema de intercomunicación por video sin costos o equipo de videoteléfono adicional.

Es un software de administración de control de acceso completo que incluye una eficiente arquitectura de sistema basada en TCP/IP con lectores IP inteligentes admite una amplia variedad de dispositivos de terceros e integraciones con otros sistemas de seguridad, lo que lo convierte en la mejor solución en cuanto a reducción de costos y flexibilidad en el diseño.

Sistema de control de acceso ExClouds

Exclouds es una aplicación basada en web del Sistema ExpansE[3]. Permite una configuración más rápida y fácil, no requiere instalación de software ni configuración de la computadora. Controla todas las opciones de configuración del sistema, así como los servicios adicionales, inclusive vigilancia en tiempo real desde cualquier panel y computadora en la red la aplicación de servidor de ExClouds complementa el sistema con capacidades avanzadas de generación de informes la interfaz web de ExpansE es fácil de usar y facilita la gestión de control de acceso, video y monitoreo de alarmas desde cualquier panel o navegador de Internet, y no se requiere operadores expertos, además la arquitectura completa de cliente-server permite el uso de múltiples-clientes a niveles diferentes de usuario.

Sistemas de control de acceso desarrollados en la Universidad de las ciencias Informáticas (UCI).

Sistema de Identificación

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

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

Sistema de Control de Acceso a Comedores

Mediante este sistema se controla en los comedores de las diferentes edificaciones donde se brinda el servicio de alimentación; el acceso de los estudiantes, profesores y trabajadores durante las tres sesiones de servicio: desayuno, almuerzo y comida. El mismo se divide en dos partes: el control de acceso y la gestión de comensales. El acceso se controla registrando el código de barras, que se encuentra en la identificación de cada persona, en cada una de las puertas de los comedores. La gestión de comensales permite a los directivos la asignación de los comedores y puertas a los mismos, además de ofrecer reportes como la cantidad de comensales desglosado por puerta o tipo (Ramírez, et al., 2015).

Sistema de control de acceso a los laboratorios de producción (UCILAB)

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

Sistema de control de acceso para el centro CISED en la UCI

El sistema se encuentra actualmente en funcionamiento, en el centro de desarrollo de software CISED, con el objetivo de controlar el personal que accede a los laboratorios asignados a la producción. La funcionalidad principal del controlador de acceso es impedir que el usuario pueda acceder a los servicios de red si este no se ha registrado por el controlador de entrada, aplicación web que recibe el número de solapín. El sistema no puede controlar las aplicaciones iniciadas por el usuario en las estaciones de trabajo (Ramírez, et al., 2015).

1.3.1. Valoración de los sistemas estudiados

Luego del estudio realizado a los sistemas de control de acceso descritos anteriormente se logró identificar varias de las funcionalidades y características comunes de estos softwares:

- El control de horarios y los permisos concedidos.
- La capacidad de generar informes de eventos de identificación, administración y estaciones de trabajo, esto es sumamente útil a la hora de extraer históricos sobre cualquier evento ocurrido en el local de trabajo.
- Se identificó la arquitectura Cliente-Servidor, la cual permite el uso de múltiples clientes a niveles diferentes de usuarios.
- El diseño para la conveniencia y simplicidad del usuario.

Los sistemas que, desde la perspectiva del autor, más se asemejan a una posible solución de la problemática planteada, debido a las características expuestas son:

- Sistema de control de acceso a los laboratorios de producción (UCILAB), existen varias implementaciones de este sistema en la UCI. Cada una de ellas específica para el área productiva donde se encuentra, lo que hace que no exista una base de datos centralizada con todos los datos referentes a todos los laboratorios de producción, no en la base de datos un personal ajeno al área productiva donde está implantado el sistema.
- Sistema de control de acceso a comedores, aunque el negocio que maneja no se ajusta a la situación polémica planteada, el estudio de dicho sistema permitió observar como ejecuta el consumo de los servicios de LDAP de la Universidad, así como modelar las funciones asociadas a visualizar el local correspondiente a cada especialista.

En el caso de los controladores de acceso internacionales como lo son “Easy Way”, “ExClouds” y “Arquero”, son sistemas propietarios con un alto costo de adquisición, además son aplicaciones personalizadas a clientes específicos en cada empresa las características expuestas permiten tener una idea de forma general de cómo solucionar la problemática en cuestión.

De los sistemas controladores de acceso internacionales se obtuvieron ideas de las funcionalidades a implementar como:

- El uso de la arquitectura Cliente-Servidor, haciendo uso de esta arquitectura está el sistema “ExClouds”.
- La capacidad de generar informes de eventos de identificación y su utilidad a la hora de extraer históricos sobre cualquier evento ocurrido, estas funcionalidades fueron implementadas por los sistemas “Easy Way” y “Arquero”.
- La seguridad en el acceso de los usuarios a datos, estaciones de trabajo y redes, por parte del sistema “Digital Persona”.

1.3.2. Tendencias y tecnologías actuales

Son diversas las tecnologías que pueden ser usadas para la elaboración de un producto de software. Tener una clara visión de las tendencias y estándares actuales en el desarrollo de aplicaciones informáticas juega un papel fundamental en la tarea de definir las tecnologías, que permitan satisfacer los requerimientos de la solución a desarrollar teniendo en cuenta sus características.

Se hace necesario centrar el análisis en las tecnologías que son usadas en la esfera de la gestión de seguridad.

Conclusión del estudio del arte.

Luego de investigar acerca de los sistemas de control de acceso que constituyen la base para la concepción, diseño e implementación de la solución propuesta, se logró identificar varias de las funcionalidades y características comunes de estos software: el control de horarios y los permisos concedidos, la capacidad de generar informes de eventos de identificación, administración y estaciones de trabajo, esto es sumamente útil a la hora de extraer históricos sobre cualquier evento ocurrido en el local de trabajo, se identificó la arquitectura Cliente-Servidor, la cual permite el uso de múltiples clientes a niveles diferentes de usuarios, el diseño para la conveniencia y simplicidad del usuario. Se decide desarrollar una nueva solución, donde, será utilizado el modelo de control de acceso basado en roles siguiendo la propuesta de que a cada usuario le asigna los derechos de acceso a un rol y los roles son relativamente estables en comparación a los usuarios.

1.5. Herramienta y tecnologías a utilizar en la investigación.

Lenguaje de modelado

Lenguaje Unificado de Modelado (**UML**), es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo, se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar. (shumller, 2014).

Herramienta de modelado (Visual Paradigm 15.1)

Visual Paradigm para UML es una herramienta de modelado UML y herramienta CASE de modelado profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Lenguaje de desarrollo

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras.

Según Vicente Trigo Aranda el conjunto de órdenes e instrucciones que se dan al ordenador para que resuelva un problema o ejecute una determinada misión, recibe el nombre de programa. En los primeros tiempos de la informática, la programación se efectuaba en el único lenguaje que entiende el microprocesador: su propio código binario, también denominado lenguaje máquina o código máquina. (Aranda, 2018).

Lenguaje PHP 7.2

PHP (Hipertexto Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML y ejecutado en el servidor con PHP no se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF (Portable Document Format), también puede presentar otros resultados, como XHTML[4] y archivos XML[5]. Así mismo, puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla. Quizás la característica más potente y destacable es su soporte para una gran cantidad de base de datos.

Lenguaje de marcas de hipertexto (HTML)

Lenguaje de marcado de hipertexto, HTML por sus siglas en inglés, hace referencia al lenguaje predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. También puede detallar, hasta un cierto punto, la apariencia de un documento. Se escribe en forma de etiquetas, rodeadas por corchetes angulares.

JavaScript

Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa (Quijano, Cleto, & Stampella, 2019).

Se resuelve usar dicho lenguaje en su versión 5.0 ya que agrega dinamismo a los portales web, así como validación de campos y formularios. No es un lenguaje compilado si no, interpretado, lo hace ideal para ejecutarse en los navegadores actuales. El uso de variables, funciones y operadores es semejante a los demás lenguajes de programación de más alto nivel. JQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones.

---

[4] XHTML: Siglas del inglés eXtensible HyperText Markup Language. XHTML es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

[5] XML *eXtensible Markup Language* es un lenguaje de marcas.

Hojas de estilo cascada (CSS)

Según Eguiluz CSS son un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla. Se utilizan para dar estilos a documentos HTML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML. El principal objetivo del uso de CSS es separar la estructura del documento de su presentación y de esta forma aumentar la organización del código y el riesgo de pérdida de uno u otro, además les ofrece a los desarrolladores el control total sobre el estilo y formato de sus documentos. Con la utilización de CSS se pueden aplicar los diferentes estilos al sistema.

Framework de desarrollo

Según Javier J. Gutiérrez, el concepto framework se emplea un mucho ámbito del desarrollo de sistemas software, no solo en el ámbito de aplicaciones web.

Framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Symfony 4.0

Symfony4.0 es la versión más reciente de Symfony, el popular framework para desarrollar aplicaciones PHP. Hay frameworks para trabajar en el lado del servidor (backend) o en el lado del cliente (frontend), o ambos. Symfony es un framework del lado del servidor.

Según Víctor Manuel Acosta Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador(MVC)**.** Esto quiere decir que, para los desarrolladores, separa la lógica de negocio, la lógica de servidor y la capa de presentación. Aunque Symfony4.0 utiliza el patrón de diseño Modelo Vista Controlador, tiene su propia forma de trabajo. En línea con MVC, los archivos de la aplicación se distribuyen en carpetas según su función la estructura que propone Symfony 4.0 es opcional, si no se requiere una carpeta no es necesario crear al directorio. Las carpetas más importantes de una aplicación realizada con Symfony, en su versión 4.0 son las siguientes:

- **Tests:** archivos de pruebas de la aplicación. Permite declarar un espacio de nombres de prueba específico para la carga automática.

- **Templates:** este directorio solamente se crea cuando se instala Twig (plantillas para php). Contiene las plantillas de nuestra aplicación.

- **Config:** contiene los archivos de configuración. Encontraremos un archivo por paquete y por entorno. También contiene un archivo para cada configuración de enrutamiento.

- **Src:** aquí se incluye la lógica de negocio. Dentro de este directorio se encuentran los controladores la aplicación.

- **Var:** incluye los archivos temporales, entre los que se encuentra la caché.

Sistema gestor de base de datos(SGBD).

Según (Marín, 2019) un sistema gestor de base de datos (SGBD) o DataBase Managenent System (DBMS) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible. En la actualidad, existen multitud de SGBD y pueden ser clasificados según la forma en que administran los datos en: relacionales(SQL), no relacionales(NoSQl).

PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de Postres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multa-versión que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación. Posee características significativas del motor de datos, robustez, eficiencia y estabilidad, multiplataforma, flexibilidad en cuanto a lenguajes de programación, dispone de una herramienta (pgAdmin) muy fácil e intuitiva para la administración de las bases de datos.

Servidor web Apache v.2.4

Apache es el servidor web que provee servicios del Protocolo de Transferencia de Hipertexto (por sus siglas en ingles HTTP). Al ser altamente configurable, robusto y estable hacen que cada vez millones de usuarios reiteren su confianza en este programa. El enfoque principal de la actualización de Apache 2.4 es la mejora en el rendimiento que se sintetizan en un menor consumo de memoria y mejoras en las concurrencias de las peticiones. Es la versión más rápida de Apache (Meloni, 2012).

Los diferentes módulos de multiprocesador disponibles en Apache 2.4 permiten a los administradores de sistemas ajustar Apache para ser más rápido según las necesidades y la naturaleza de las peticiones que tenga que atender. Estos módulos pueden ser seleccionados en tiempo de ejecución con lo que añade una mayor flexibilidad. Incluso presume de tener un rendimiento superior a los servidores orientados a eventos (Meloni, 2012).

Entorno de desarrollo integrado

Habitualmente cuentan con una avanzada interfaz gráfica de usuario (GUI). Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación que puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity). Cuenta con muchas características como Intellisense, Git, snippets, y es customizable por lo que sus usuarios pueden cambiar las preferencias, shortcuts y temas a su gusto.

1.6. Metodología de desarrollo de software

Desde el punto de vista informático una metodología de desarrollo de software es el conjunto de procedimientos, técnicas, herramientas y un soporte documental a la hora de desarrollar un producto de software, que indica quién, cuándo y cómo hacer algo. La presencia de una metodología en el desarrollo de un proyecto garantiza un producto con más calidad y reduce el tiempo de entrega del producto ya que al tener una mejor planificación la producción permite cumplir con la fecha establecida (Arias, 2013). Las metodologías de desarrollo se clasifican en dos grupos; las tradicionales o robustas y las ágiles, a continuación, se explican en qué consiste cada una de ellas: metodologías robustas o tradicionales y metodologías ágiles.

Metodologías ágiles

Su objetivo es esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente, respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretende ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. (Rodríguez, et al., 2008).

Principales características de las metodologías ágiles

- Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.
- Se hace mucho más importante crear un producto software que funcione que generar mucha documentación.

- El cliente está en todo momento colaborando en el proyecto. (Rodríguez, et al., 2008).

Visto esto se puede concluir que, en el desarrollo de software es importante la selección de una adecuada metodología para lograr el éxito del producto final. A continuación, se describe la metodología seleccionada para guiar el proceso de desarrollo de software de la presente investigación.

AUP-UCI es una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. En aras de aumentar la calidad del software que se produce esta metodología se apoya en el Modelo CMMI-DEV v1.3. (por sus siglas en inglés Capability Maturity Model Integration Development, Integración de Modelos de Madurez de Capacidades para Desarrollo). El mismo constituye una guía para aplicar las mejores prácticas y obtener un producto o servicio con calidad en una entidad desarrolladora. (Sánchez, 2015).

Descripción de las Fases

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Inicio Ejecución y se agrega la fase de Cierre.

- **Inicio:** El objetivo de esta fase es llevar a cabo las actividades relacionadas con la planeación del proyecto. Se realiza un estudio inicial de la organización que actúa como cliente y se obtiene información clave acerca del alcance del proyecto, se realizan estimaciones de tiempo y esfuerzo, y finalmente se decide si se ejecuta o no.
- **Ejecución:** En esta etapa se ejecutan las actividades requeridas para desarrollar el software. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elabora la arquitectura y el diseño, se implementa y se libera el producto.

- **Cierre**: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se llevan a cabo las actividades formales de cierre de proyecto.

La metodología AUP-UCI se ajusta al ambiente que presenta el negocio que se desea informatizar y da la posibilidad al cliente de siempre acompañar al equipo de desarrollo para convenir los requisitos y así poder implementarlos. También es una de las que mayor prestigio presenta hoy en día entre los desarrolladores de la UCI, pues fue aprobada cuando la institución se sometió al proceso de certificación CMMI nivel 2 para el desarrollo de software. Además, es una metodología que se adapta a cualquier proyecto productivo de la UCI, y el artefacto que vamos utilizar para mostrar evidencia del proceso de desarrollo, es el modelado de negocio basado en el caso de uso del negocio (CUN).

1.7. Conclusiones del capítulo

En este capítulo se analizaron las condiciones y problemas que rodean al objeto de estudio y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que tributan al problema y sobre la base de ese se obtuvieron los objetivos generales y específicos para este trabajo, se seleccionaron además las herramientas y la metodología de desarrollo que constituían la opción más viable para dar solución a dicha aplicación.

Capítulo 2: Análisis de la propuesta de solución

Introducción

En este capítulo se describe las principales características de la solución propuesta a través del modelo de dominio y una descripción de los procesos que serán objeto de automatización proporcionando un mejor entendimiento del sistema. Se hace un levantamiento de los requisitos funcionales y no funcionales necesarios para lograr que el sistema web funcione correctamente. Además, se construyen los artefactos correspondientes al análisis y diseño acorde con la metodología seleccionada en el capítulo anterior. También, los patrones de la arquitectura a utilizar y diagramas de diseño. Luego se definen los diagramas de componentes y despliegue, los cuales tienen un papel fundamental en la etapa de construcción del sistema.

2.1. Modelo de dominio

Un Modelo Conceptual o Modelo de Dominio, constituye una representación visual para el usuario de los conceptos u objetos significativos del mundo real para un problema o área de interés. Representa conceptos del mundo real, no de los componentes de software, mediante clases conceptuales del dominio del problema, encargándose de capturar los tipos más importantes de objetos y eventos que suceden en el entorno (Pressman, et al., 2003).

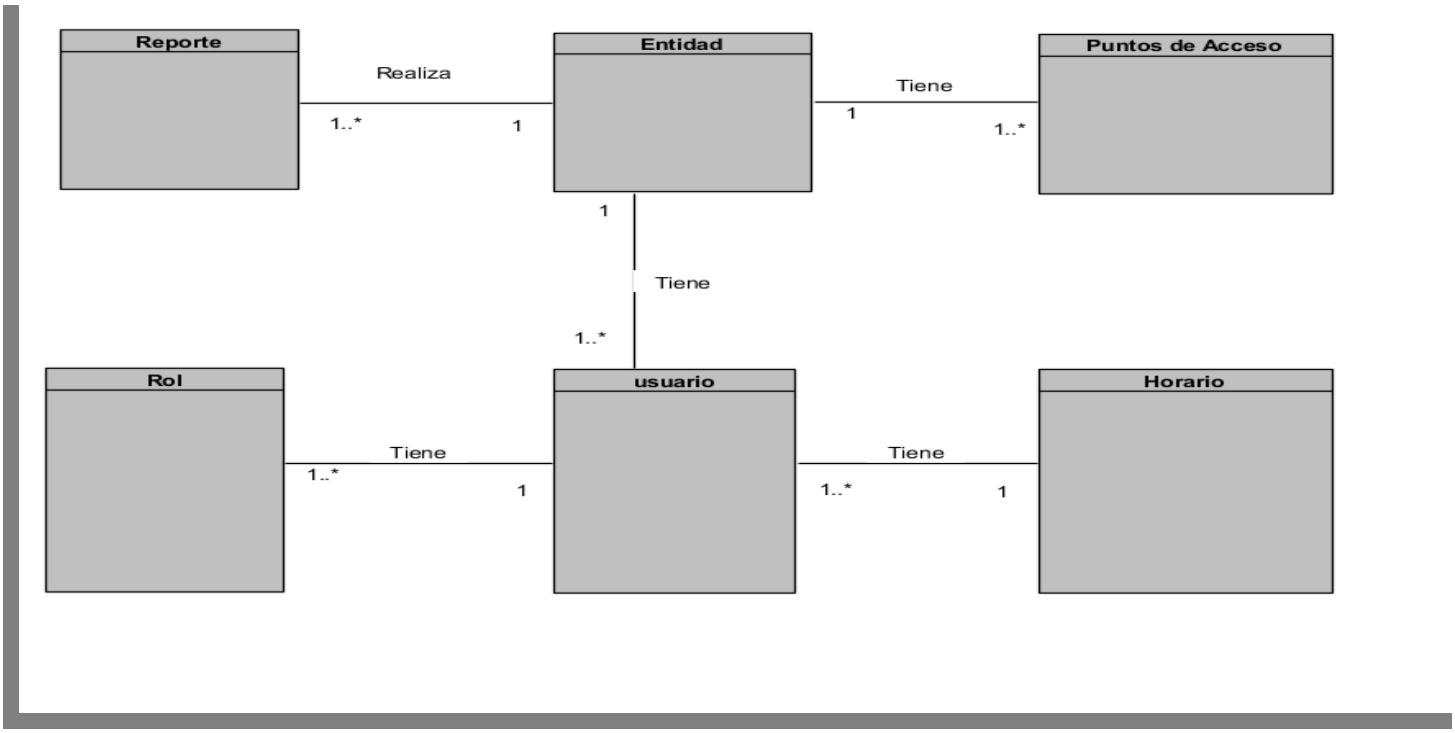


Figura 1: Modelo de dominio

2.1.1. Descripción de los conceptos del dominio

A continuación, se describen las clases del modelo de dominio, para su mejor comprensión:

Usuario: persona que una vez registrada en el sistema en correspondencia al rol que tenga asignado, puede acceder a los puntos de accesos que está autorizado.
Rol: función que cumple determinado usuario en el sistema.
Horario: horario de producción de los usuarios (trabajadores o visitantes), de la institución está dividido entre los días de la semana y las secciones de trabajo.
Puntos de acceso: entidad que realiza el control de todos los lugares que el usuario tiene acceso.
Entidad: representa la estación de trabajo o la institución que se desea acceder.
Reportes: entidad que representa las incidencias que son realizadas una vez que la entidad(institución) procesa el control del comportamiento de acceso por horario.

2.1.2. Descripción de la propuesta de solución

Como solución a la problemática descrita en la introducción del presente trabajo se propone la implementación de un sistema web para el control de acceso en entidades que permitirá informatizar el proceso de control de acceso físico, este sistema estará compuesto por 4 módulos a petición del cliente: Administración, Configuración, Reportes y Control de acceso. En la presente investigación se implementarán cada uno de ellos. Dicho sistema mantendrá el control de las entradas y salidas del personal, así como el acceso a las estaciones de trabajo en cuanto a sesiones. Además, de generar las incidencias que ocurran durante la estancia de un determinado usuario en la estación de trabajo.

El módulo Administración se encarga de gestionar todos los datos personales de un determinado usuario, los roles que cada uno tenga en la institución y los grupos de roles asociado al usuario, el sistema tiene un tipo de rol, el cual será asignado a un tipo de usuario principal, que tendrá acceso al sistema: administrador, este es el encargado de controlar el acceso en la entidad. El primer rol cuenta con permisos globales lo cual posibilita acceder y ejecutar cualquier funcionalidad o módulo del sistema y por último mostrará la información de la funcionalidad que se ejecutará en el momento.

El módulo configuración se encarga de gestionar todos los puntos de acceso de la institución, nos permitirá configurar los horarios de entrada y salida de cada usuario en la institución, las personas circuladas y las causas de una posible violación en la institución y también permitirá configurar la credencial para cualquier tipo de entidad.

El módulo control de acceso se encargará de autorizar o denegar un usuario que necesita acceder al sistema, podrá controlar todos los usuarios que están autenticado en el sistema y mostrará una lista de estadísticas de acceso, lista del último acceso, los datos del punto, así como buscar una persona y mostrará los datos de una persona accedida.

El módulo Reportes brinda el servicio de generar reportes y exportarlos en formato pdf en este módulo se tendrá un listado de todos los accesos en la entidad así como una lista de todas las personas circuladas en la institución permitiendo tener un control de comportamiento de acceso por horario de forma gráfica.

2.2. Requisitos Funcionales

Los requisitos funcionales definen el comportamiento interno de un software, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software. (SOMMERVILLE, 2007).

Tabla 2: Requisitos funcionales.

Módulos	Requisitos
Control de acceso	RF 1. Autenticar. RF 2. Mostrar escritorio. RF 3. Mostrar lista de ultimo acceso. RF 4. Mostrar datos del punto. RF 5. Mostrar estadísticas de acceso. RF 6. Buscar Personas. RF 7. Mostrar datos de persona accedido.
Administración	RF 8. Crear usuario. RF 9. Modificar usuario. RF 10. Activar usuario. RF 11. Desactivar usuario. RF 12. Ver detalle de usuario. RF 13. Eliminar usuario. RF 14. Crear rol. RF 15. Ver detalle de rol. RF 16. Modificar rol. RF 17. Eliminar rol. RF 18. Registrar grupo de rol. RF 19. Listar grupo de rol. RF 20. Modificar grupo de rol. RF 21. Eliminar grupo de rol. RF 22. Cambiar contraseña.
Reportes	RF 23. Exportar listado de accesos. RF 24. Listar accesos. RF 25. Exportar lista de personas circulada. RF 26. Listar personas circulada. RF 27. Graficar comportamiento de acceso por horario.

Configuración	RF 28. Crear institución. RF 29. Modificar institución. RF 30. Activar institución. RF 31. Desactivar institución. RF 32. Ver detalle de la institución. RF 33. Crear horario. RF 34. Modificar horario. RF 35. Activar horario. RF 36. Desactivar horario. RF 37. Ver detalle del horario. RF 38. Crear puntos de acceso. RF 39. Modificar puntos de acceso. RF 40. Activar punto de acceso. RF 41.ver detalle de los puntos de accesos. RF 42. Listar puntos de accesos RF 43. Circular persona RF 44. Crear causa RF 45. Modificar causa RF 46. Listar causa. RF 47. Eliminar causa. RF 48. Crear credencial. RF 49. Eliminar credencial. RF 50. Exportar credencial

2.3.2. Requisitos no funcionales

De acuerdo con (SOMMERVILLE, 2007), los requisitos no funcionales son aquellos requisitos que no describen información a guardar, ni funciones a realizar, sino que son propiedades que hacen al producto atractivo, usable, rápido o confiable. Además, se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software.

Usabilidad

- El sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo por parte de los usuarios, los cuales deben poder acceder de manera rápida y efectiva a la información solicitada. Debe además ser una interfaz de manejo cómodo donde la curva de aprendizaje para los usuarios sea lo menos inclinada posible y que posibilite en estos una rápida adaptación..
- El sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por su parte sobre la herramienta de trabajo.

Fiabilidad:

- El sistema estará disponible las 24 horas del día y los siete días de la semana..

Rendimiento

- Se tiene en cuenta que el producto se debe diseñar sobre una arquitectura cliente - servidor, los tiempos de respuestas del sistema deben ser rápidos: no debe de superar los 5 segundos.

Seguridad

- El acceso a la aplicación se realizará a través del modelo RBAC (Control de Acceso Basado en Roles), teniendo solo acceso a esta el personal autorizado y solo a las funcionalidades definidas.
- Confiabilidad: la información que se maneje en el sistema estará protegida de acceso no autorizado y divulgación, a partir de los diferentes roles de los usuarios que empleen el sistema.
- Disponibilidad: la información se encontrará disponible en todo momento para aquellos usuarios autorizados a acceder al sistema.
- Solo el usuario previamente autenticado podrá hacer uso del sistema.

Requisitos del software

- El servidor deberá contar además con la instalación de Gestor de Bases de Datos PostgreSQL y el servidor web apache.

Restricciones de diseño

- Lenguaje de programación: PHP 7.2 o superior

- El marco de trabajo base de desarrollo que se utilizará es: Symfony 4.x
- Como IDE se empleará Visual Studio Code
- El sistema gestor de bases de datos deberá ser PostgreSQL

Interfaz

- Será diseñada para adaptarse a la resolución de pantalla del usuario, utilizando colores refrescantes y agradables.
- Interfaz Web: La interfaz deberá ser sencilla con tonalidades azules claras y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.

2.4. Modelo de casos de uso del sistema

Los componentes primarios de un modelo de casos de uso son los casos de uso, los actores y el sistema modelado. Un caso de uso es una técnica de modelado usada para describir lo que debería hacer un sistema nuevo o lo que hace un sistema que ya existe. En el modelado de casos de uso, el sistema se observa como una caja negra que proporciona casos de uso. Cómo lo haga el sistema, cómo se implementen los casos de uso y cómo trabajen internamente no importa. (Mediavilla, 2014).

2.4.1 Descripción de los actores del sistema

Después de haberse definido los requisitos funcionales y no funcionales que el sistema debe cumplir, corresponde definir los actores que intervendrán en el sistema.

Actores del sistema: Un actor o algo que interactúa con el sistema, pero que es externo al sistema. (Mediavilla, 2014).

Tabla 3: Descripción de los actores del sistema.

Actores	Descripción
Usuario del Sistema	Persona que accede al sistema para registrar su asistencia, visualizar su reporte o su horario de producción.
Administrador del sistema	Usuario que accede al sistema para visualizar el reporte de todos los involucrados en el sistema, además son las personas facultadas para la gestión del sistema en general, además de crear la lista de accesos a todos los puntos de la entidad.

2.4.2. Patrones de caso de uso

Los patrones de casos de uso son artefactos narrativos que describen bajo la forma de acciones y reacciones el comportamiento del sistema desde el punto de vista del usuario. Cada CU proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. (Pressman, et al., 2007).

Patrón de Caso de Uso utilizado

En el diseño de los CU del sistema se utilizó el patrón CRUD por sus siglas en inglés (Creating, Reading, Updating, Deleting), el patrón CRUD Completo consiste en un CU para administrar la información, concretamente se precisa en el: gestionar rol y gestionar control de acceso. CRUD Parcial radica en que puede ser modelada como CU independiente, se puede evidenciaren el: gestionar causa, gestionar credenciales, listar personas circuladas, lista de acceso.

Se puso en práctica además el patrón de CU actores múltiples, utilizado en el sistema para gestionar institución, gestionar roles, generar reportes, gestionar personas, autenticar usuario, listar causa, buscar personas circuladas.

Generalización/Especialización entre actores: una relación de generalización de una clase hija de actor a otra clase padre de actor indica que el hijo hereda el rol que la clase padre puede jugar respecto a un caso de uso. Se ve reflejado en el Usuario que representa la generalización de los actores: Administrador del sistema.

2.4.3. Diagrama de casos de uso del sistema(DCUS)

Un Caso de Uso (CUS) es un fragmento de una funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Estos se forman agrupando requisitos funcionales y especifican las acciones que tienen lugar durante la interacción actor-sistema. Un DCUS representa gráficamente a los procesos y sus interacciones con los actores. En la fig.4 se representa el DCUS del Sistema de Control de acceso en entidades.

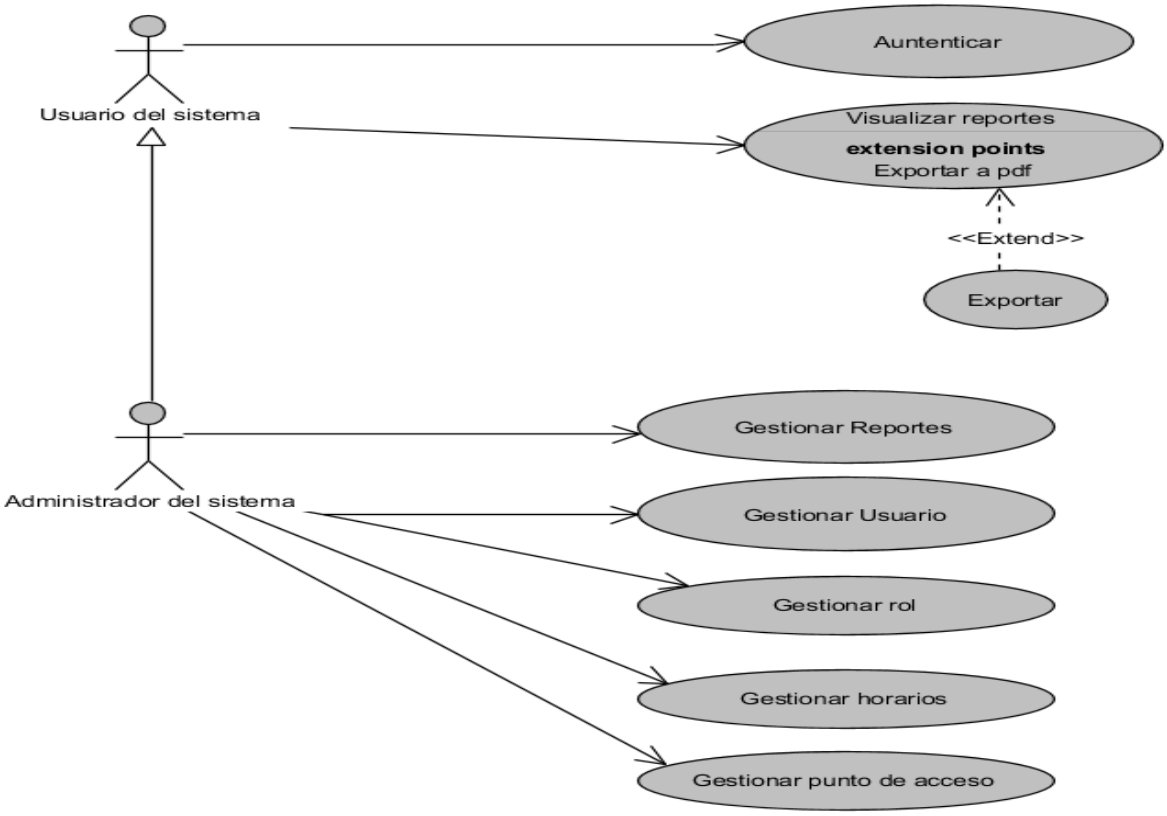


Fig. 4 Diagrama de Caso de Uso del Sistema

Descripción textual de casos de uso del sistema

Tabla 5: Descripción del CU Gestionar institución.

Caso de Uso	Gestionar institución
Actores	Usuario
Propósito	Este caso de uso se realiza con el objetivo de crear una nueva institución en el sistema.
Resumen	El caso de uso se inicia cuando el usuario accede a la tabla de contenidos en el panel derecho destinado a los módulos principales del sistema, con el objetivo de adicionar una nueva institución a la base de datos del sistema.
Precondiciones	El usuario debe estar autenticado en el sistema como administrador.
Referencias	RF 28.
Prioridad	Alta
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Se accede a la tabla de contenidos en el panel derecho. 2. Se accede al módulo Configuración. 3. Se presiona la opción institución¨. 5. Se ingresa los datos.	4. Se muestra una interfaz con los campos de los datos a ingresar. 6. Se crea la institución, y muestra una interfaz con los detalles de la institución.
Flujos alternos	
Acción del Actor	**Respuesta del Sistema**
1. Se accede a la tabla de contenidos en el panel derecho. 2. Se accede al módulo configuración. 3. Se presiona la opción institución. 5. Se ingresa los datos. 7. Se ingresa nuevos datos.	4. Se muestra una interfaz con los campos de los datos a ingresar. 6. Se muestra un mensaje, ¨ **Ya existe una institución con nombre XX. El nombre de la institución debe ser único** ¨. 8. Se crea la institución.
Poscondiciones	Se ha añadido una nueva institución en la base de datos del sistema.

Tabla 6: Descripción del CU Eliminar rol.

Caso de Uso	Eliminar rol
Actores	Usuario

Propósito	Este caso de uso se realiza con el objetivo de eliminar los roles existentes en la base de datos del sistema.
Resumen	El caso de uso se inicia cuando el usuario accede a la tabla de contenidos en el panel derecho destinado a los módulos principales del sistema, con el objetivo de eliminar el o los roles existentes en la base de datos del sistema.
Precondiciones	1. El usuario debe estar autenticado en el sistema como administrador. 2. Debe de existir en la base de datos del sistema una lista con el nombre de los roles existentes.
Referencias	RF 17.
Prioridad	Alta
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**

1. Se accede a la tabla de contenidos en el panel derecho. 2. Se accede al módulo ¨ Administración¨. 3. Se presiona el botón ¨Roles¨. 5. El usuario presiona opción de ¨Eliminar¨ en la interfaz mostrada por el sistema. 7. Se seleccionará la opción de aceptar o de cancelar. 8.1. Si se selecciona la opción aceptar. 8.2. Si se selecciona la opción cancelar.	4. Se muestra una interfaz con una lista de roles existentes en la base de datos del sistema y los detalles de los mismos. 6. Se muestra un mensaje ¨ ¿Estás seguro que desea eliminar este elemento? ¨. 9.1. Se guardará los cambios. 9.2. Se muestra una interfaz con una lista de roles existentes en la base de datos del sistema y los detalles de los mismos.
Flujos alternos	
Acción del Actor	**Respuesta del Sistema**
1. Se accede a la tabla de contenidos en el panel derecho. 2. Se accede al módulo ¨Administración¨. 3. Se presiona el botón ¨Roles ¨.	4. Se muestra una interfaz con una lista de los roles existentes en la base de datos del sistema y los detalles de los mismos.
5. Se presiona opción de ¨Eliminar¨ en la interfaz mostrada por el sistema. 7. Se debe de autenticarse en el sistema como administrador.	6. Se muestra un mensaje, diciendo que no tiene los permisos para realizar esta acción.
Poscondiciones	Se ha eliminado el o los roles definidos por el usuario.

2.4.4. Diseño de la propuesta de solución

La actividad de diseño de software se refiere al establecimiento de las estructuras de datos y la arquitectura general del software, de manera que se satisfagan los requerimientos del software. El proceso de diseño traduce los requisitos en una representación de software (Pressman, 2010).

2.5. Patrones de diseño y estilo de arquitectura

Los patrones de arquitectura son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un grupo de restricciones sobre cómo pueden ser usados. Brindan una descripción de los elementos y la relación entre ellos, junto a un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor (Pressman, 2010).

2.5.1 Patrón modelo vista controlador (Model View Controller, MVC)

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema (Larman, 2015).

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

- Es posible tener diferentes vistas para un mismo modelo.
- Es posible construir nuevas vistas sin necesidad de modificar el modelo inferior.
- Proporciona un mecanismo de configuración a componentes complejos mucho más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción).

2.5.2. Arquitectura n-capas

La arquitectura N-Capas es un estilo arquitectónico donde el objetivo principal es separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio y mecanismo de almacenamiento. Una de las ventajas principales de esta arquitectura es que el desarrollo se puede utilizar en varios niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles. El diseño más utilizado actualmente es el conocido como diseño en tres capas.

1. Capa de Presentación: Es la capa encargada de presentar el sistema al usuario, le comunica la información y captura a su vez la información introducida. Es conocida además como interfaz gráfica y se comunica con la capa de negocio.

2. Capa de Negocio: En ella residen los programas que se ejecutan, se encarga en recibir las peticiones de los usuarios y envía las respuestas tras el proceso. Aquí es donde se establecen todas las reglas que deben cumplirse. Recibe las solicitudes de la capa de presentación y presenta los resultados a la misma. Esta capa se comunica además con la capa de datos.

3. Capa de Acceso a Datos: Es la encargada de almacenar los datos y acceder a los mismos. Recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (Rodríguez, et al., 2008).

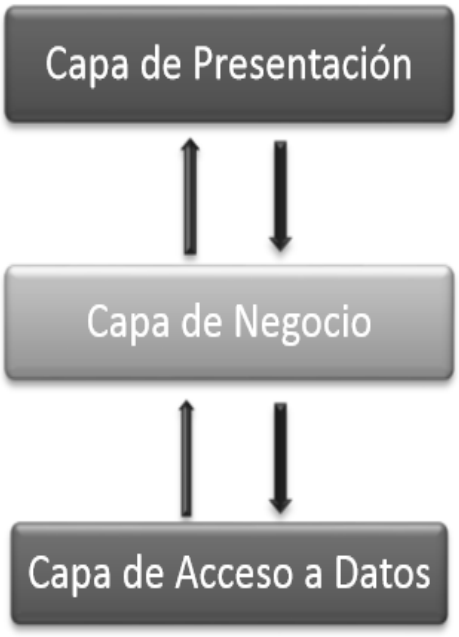


Figura 5: Arquitectura N-Capas

2.5.3. Patrones de diseño GRASP (Patrones para Asignar Responsabilidades)

Los Patrones GRASP (General Responsibility Assignment Software Patterns, o Patrones Generales de Software para Asignación de Responsabilidades), Describen los principios fundamentales de diseño para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño y el razonamiento para el diseño de una forma sistemática, racional y explicable (Andres, 2016).
Los patrones GRASP utilizados en la solución propuesta fueron:
Patrón Creador: consiste en asignar a un objeto la responsabilidad de crear otro objeto. ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello.

Patrón Controlador: sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a la vez tener un mayor control. Define además el método de su operación. Este patrón asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. En la solución, este patrón se utiliza en la clase generarReportes.

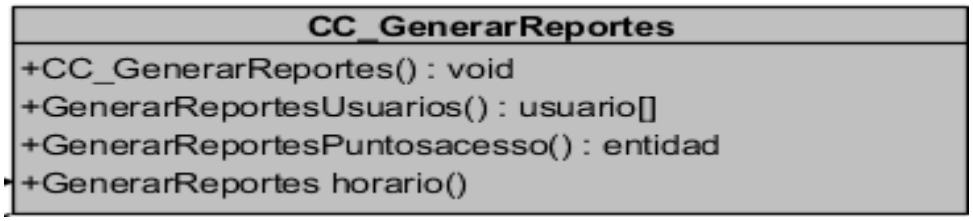


Figura 6: clase controladora generar reportes.

Patrón Experto: es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con la utilización de este patrón se facilita el encapsulamiento de funcionalidades, ya que los objetos se valen de su propia información para hacer lo que se les pide. Una de las clases en las que se utilizó el patrón experto es en la clase conexión pues cuenta con los elementos necesarios para el manejo de los perfiles y permisos que estos tienen asociados.

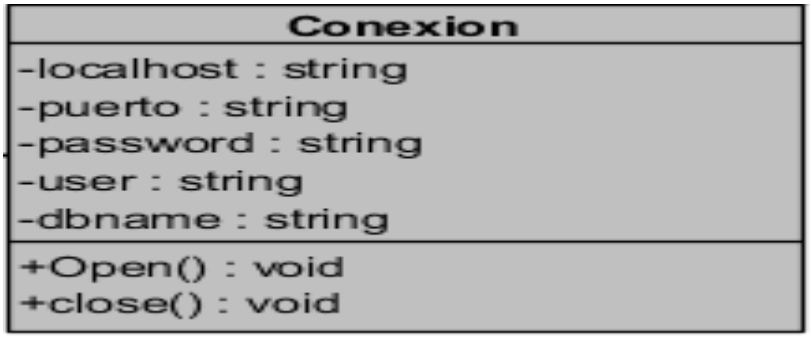


Figura 7: clase conexión generar reportes.

2.5.4. Diagrama de clases de diseño

Un diagrama de clases de diseño representa las especificaciones de las clases e interfaces software en una aplicación. Estos sirven para especificar, documentar y visualizar modelos estructurales, así como para construir sistemas ejecutables. (Pressman, 2010).

A continuación, se presenta el diagrama de clases de diseño correspondiente al CU gestionar reportes.

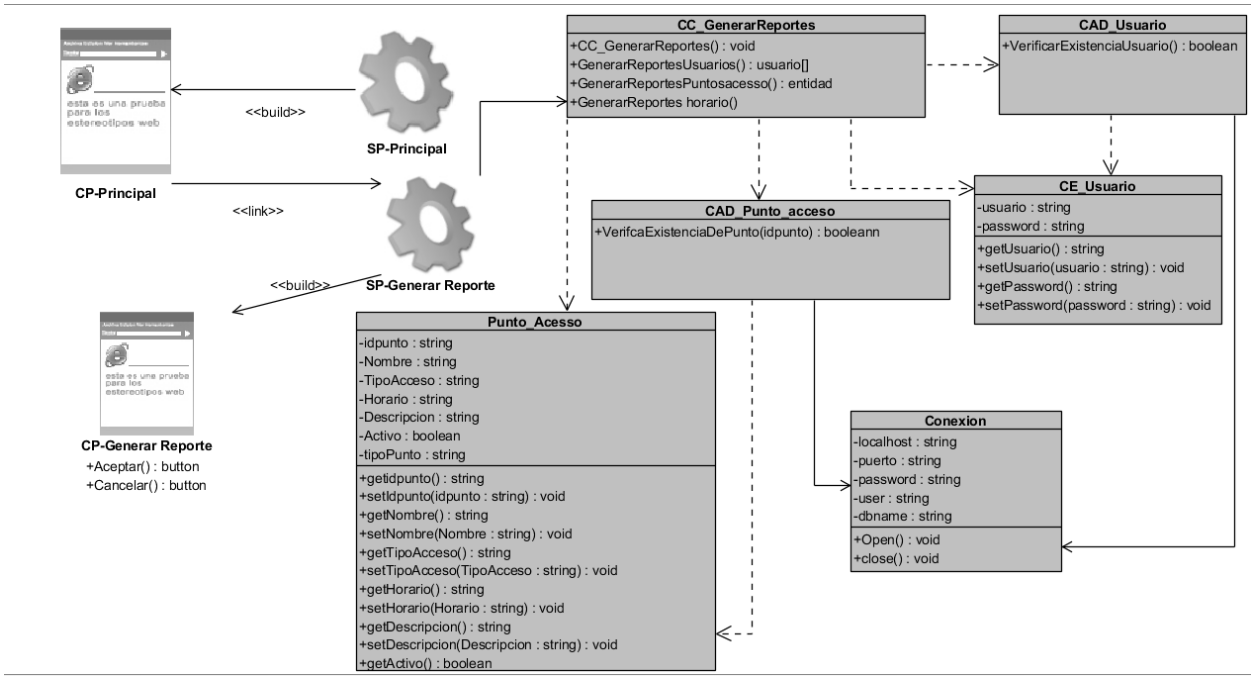


Figura 8: Diagrama de clases de diseño CU gestionar acceso.

2.5.4. Diagrama de secuencia

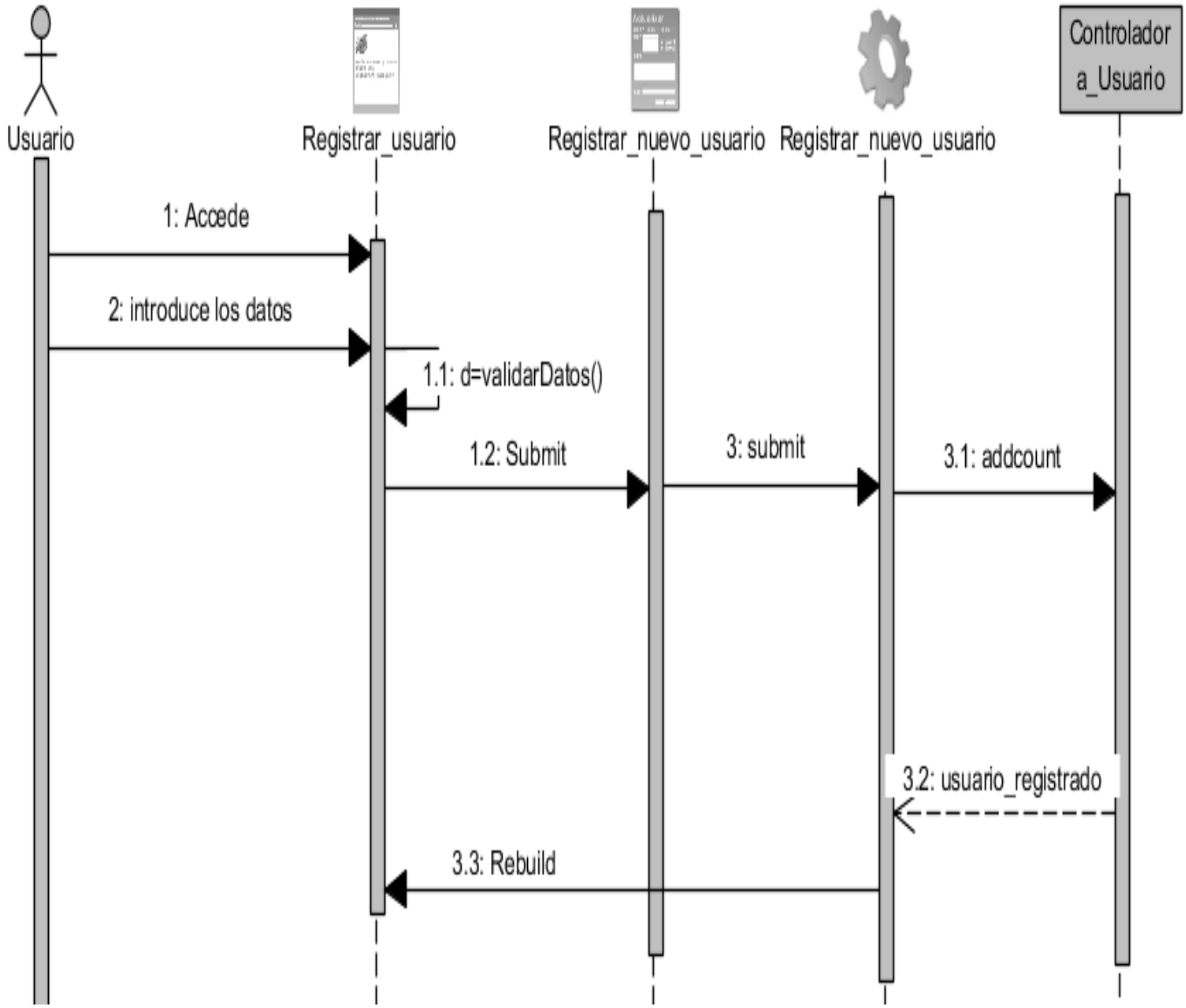
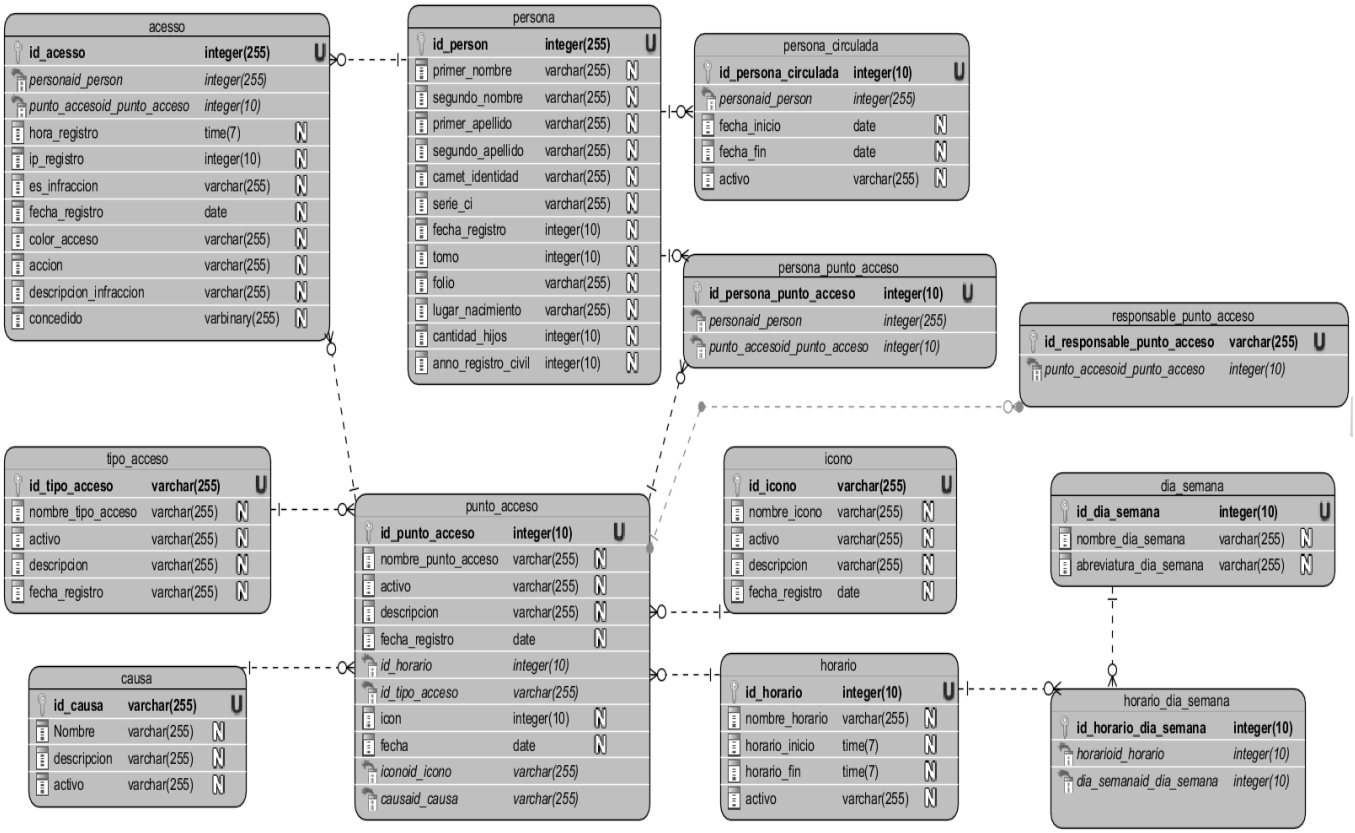


Figura 9: Diagrama de secuencia del CU gestionar usuario.

2.6. Modelo de datos

Un modelo de datos es un conjunto de reglas y convenciones bien definidas que permite aplicar abstracciones con el fin de manipular los datos del mundo real que se desean almacenar en una base de datos (SOMMERVILLE, 2007).

Figura 10: Modelo de datos.



Descripción de las tablas del modelo de datos: **Institución:** Esta tabla del modelo de datos almacena el nombre que es único, la categoría de la credencial, si está activo o no y una descripción de cada institución, cada institución tiene asignado uno o varios usuarios y uno o varios puntos de accesos, y una institución puede tener uno o varios reportes. La prioridad de esta tabla es media.

Persona: en esta tabla se almacenan todos los datos del usuario que pueden ser un administrador o mismo un usuario de sistema, que tienen diversos roles y tienen autorización para acceder a los espacios reales de la institución, el atributo credencial es único porque cada usuario tiene un numero de credencial que es su identificador. La prioridad de esta tabla es media.

Puntos de accesos: en esta tabla se almacena todos los puntos de accesos de cada institución y tiene asignado uno o varios horarios del día en que se accede a la institución y el tipo de acceso que puede ser de entrada o de salida lo cual queda reflejado en el atributo tipo. La prioridad para el sistema que ofrece esta entidad es alta.

Horario: en esta tabla horario se guarda el momento exacto en que un administrador o un visitante realiza un acceso a la institución o el momento en que ocurre un incidente. Por cada acceso, se agrega una nueva tupla a la entidad, dado que estos pueden ser realizados en cualquier hora del día.

VisitanteAcess: esta tabla representa cualquier individuo que no forma parte de la organización y desee visitar la misma, estos visitantes pueden ser trabajadores internos o personal externo de la institución, si el visitante es externo a la institución los campos categoría, cargo e imagen quedan en blanco, en el atributo credencial queda almacenado el número de Identidad del carnet del visitante en el caso de que el visitante sea un trabajador interno de la institución, en los campos antes mencionados se almacenarían los datos del usuario. El campo credencial es único para cada visitante que realiza un acceso a la institución. La prioridad para el sistema de esta tabla es media.

2.6. Conclusiones Parciales

Una vez culminadas las actividades de planificación y diseño de la propuesta de solución se obtuvo una visión más clara y detallada del sistema que se desea desarrollar. Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de requisitos, constituyeron elementos claves en la construcción de la propuesta de solución, además, se modeló la base de datos de la aplicación para hacer una representación gráfica de su estructura así especificar cómo será construida y utilizada.

Capítulo 3: Implementación y pruebas

Introducción

En el presente capítulo se muestran los artefactos correspondientes a las etapas de implementación los diagramas de componentes y el diagrama de despliegue, así como los estándares de codificación que debe seguir el equipo de desarrollo para implementar el software, de acuerdo a la metodología que se utiliza, se especifican los tipos de pruebas que esta plantea así como las que serán empleadas para validar la propuesta de solución.

3.1 Diagrama de componentes

Un diagrama de componentes muestra cómo se organizan los elementos del sistema y las dependencias existentes entre ello, permiten visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces.

Un diagrama de componentes muestra las dependencias lógicas entre componentes de software y su relación con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones; pero un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clases, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. (Jiménez, 2016).

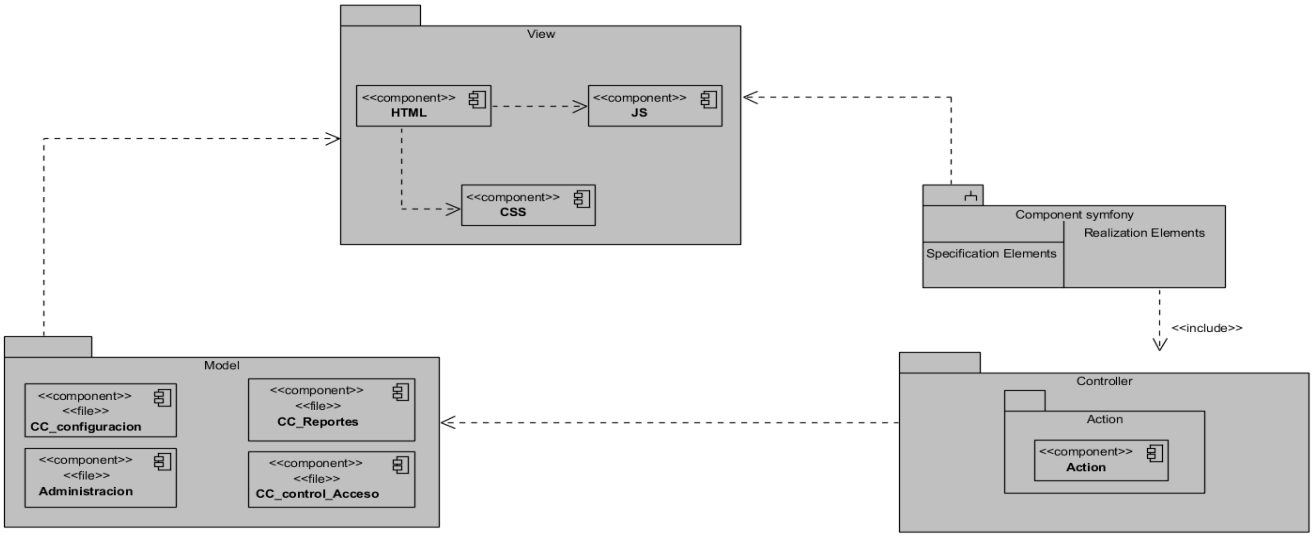


Figura 11: Diagrama de componentes del sistema.

Descripción general del diagrama de componente del sistema.

Cuando el usuario solicita ver una interfaz del sitio, internamente sucede lo siguiente: El sistema de enrutamiento determina qué controlador está asociado con la interfaz. Symfony ejecuta el controlador asociado a la interfaz, un controlador es una clase PHP en la que puedes ejecutar cualquier código que quieras. El controlador solicita al modelo los datos de la entidad. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos. Con los datos devueltos por el modelo, el controlador solicita a la vista que cree una interfaz mediante una plantilla y que inserte los datos del modelo el controlador entrega al servidor la interfaz creada por la vista.

Controller

El controlador es la parte de la aplicación que contiene lo que se llama la lógica de negocio, que es una forma elegante de decir que cada controlador se encarga de una funcionalidad completa de la aplicación.

Actions

Las acciones son el núcleo de la aplicación, puesto que contienen toda la lógica de la aplicación. Las acciones realizan llamadas al modelo y definen variables para la vista. Cuando se realiza una petición web en una aplicación Symfony4, la URL define una acción y los parámetros de la petición.

Modelo

Paquete que agrupa las clases que representan el dominio de entidades de la base de datos y que permiten la interacción directa con el paquete Vista.

View

La vista se encarga de producir las interfaces que se muestran como resultado de las acciones. Paquete que agrupa a todos los componentes que interactúan con el paquete de clases Modelo; estos componentes permiten trabajar con algunas utilidades sobre los formularios y la renderización de la información.

3.2. Diagrama de despliegue

El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Permite el mapeo de procesos dentro de los nodos, asegurando la distribución del comportamiento a través de aquellos que son representados (Pressman, 2010).

El diagrama de despliegue es un tipo de diagrama del lenguaje unificado de modelado que muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos. Se utilizan para modelar la vista de despliegue estática de un sistema, lo que implica modelar la topología del hardware sobre el que se ejecuta. Los elementos usados por este diagrama son:

- Nodos o elementos de procesamiento con al menos un procesador.
- Componentes caracterizados por ser nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.

Asociaciones que expresan el tipo de protocolo utilizado entre el resto de los elementos del modelo. A continuación se presenta el diagrama de despliegue que se propone para el sistema.

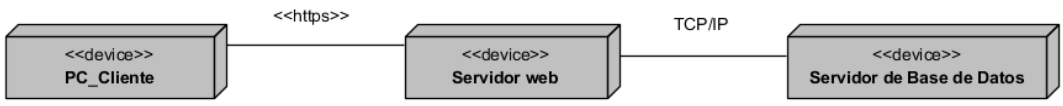


Figura 12: Diagrama de despliegue.

A continuación, la descripción de los nodos presentes en el diagrama:

- **PC Cliente:** representa el conjunto de computadoras a través de las cuales los usuarios pueden actualizar y consultar la información que se encuentra en el Servidor web. Para acceder al sistema, las PC Clientes utilizan un navegador web, la comunicación entre las PC Cliente y el Servidor web se establece utilizando el conjunto de protocolos de comunicación HTTPS (Protocolo seguro de transferencia de hipertexto).
- **Servidor web:** se ubican las capas de presentación, lógica del negocio y de acceso a datos del sistema, así como los servicios que se brindan. Este nodo representa el servidor donde está alojada la aplicación web destinada al control administrativo.
- **Servidor de Base de Datos:** representa un servidor PostgreSQL, en el cual se ubica toda la información persistente del sistema, almacenándose los datos que son utilizados y consultados por los usuarios del sistema.

Descripción del tipo de comunicación:

- La comunicación entre el nodo que representa la PC Cliente y el nodo Servidor Base de Datos se realiza mediante el protocolo TCP/IP por el puerto 5432.
- El nodo Servidor Web establece comunicación con el Servidor Base de Datos mediante el protocolo TCP/IP por el puerto 5432.

3.3. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación del código fuente de un software. Es preciso definir una serie de pautas que lleven por objetivo uniformar la estructura del código de manera tal que este sea lo más legible posible, como si un único programador hubiese escrito todo el código de una sola vez (Microsoft, 2015). Las pautas fundamentales definidas para la implementación son las siguientes:

- El tamaño máximo de las líneas de código debe ser de cien a ciento veinte caracteres aproximadamente, de manera tal que se garantice la completa visibilidad de las líneas de código sin necesidad de realizar desplazamiento horizontal.

```php
<?php

namespace App\Controller;

use App\Entity\Causa;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

/**
 * @Route("/causa", name="causa_")
 */
class CausaController extends AbstractController
{
    /**
     * @Route("/", name="listar", methods={"GET"})
     */
    public function listar()
    {
        $causas=$this-> getDoctrine()->getRepository( persistentObject: Causa::class)->findAll();
        return $this->json([
            'causas' => $causas

        ]);
    }
```

Figura 13: Fragmento de código de la clase CausaController.php.

3.4. Pruebas

Una vez generado el código fuente es necesario comenzar las pruebas del sistema para descubrir y corregir la mayor cantidad de errores posibles antes de entregarlo al cliente. Su objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores. Para ello se utilizan las técnicas de prueba que tengan una alta probabilidad de comprobar la lógica interna, las interfaces, las colaboraciones entre los componentes y los requisitos externos (Pressman, 2010).

Cualquier producto de ingeniería puede probarse de una de estas dos formas:

- Conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y al mismo tiempo buscando errores en cada función.
- Conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

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

3.4.1. Pruebas de caja blanca

La prueba de caja blanca es un método de diseño de casos de prueba que evalúa el comportamiento interno del software y usa la estructura de control del diseño procedimental para derivar los casos de prueba. En ella se realiza un examen minucioso de los detalles procedimentales, comprobando los caminos lógicos del programa, comprobando los bucles y condiciones, y examinado el estado del sistema en varios puntos. (Pressman, 2010).

Existen varios tipos de pruebas de caja blanca entre los cuales se encuentran los siguientes:

- Prueba del camino básico: permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Cualquier representación del diseño procedimental se puede traducir a un grafo de flujo.
- Prueba de condiciones: consiste en el diseño de los casos de prueba donde se ejercitan las condiciones lógicas en el módulo de un programa. Se basa en el criterio de que, si un conjunto de pruebas de un programa P es efectivo para detectar errores en las condiciones que se encuentran en P, es probable que el conjunto de pruebas sea también efectivo para detectar otros errores en el programa P.

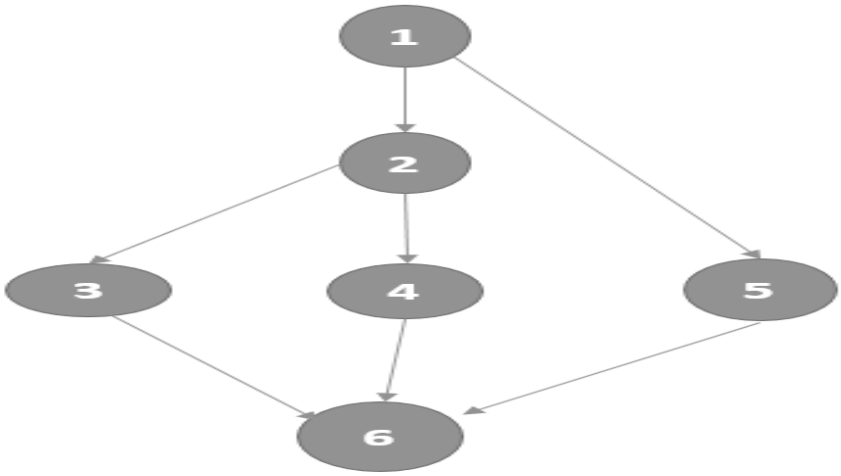
En la solución propuesta, se aplicó la técnica de pruebas de caja blanca del camino básico, puesto que permite obtener una medida de la complejidad lógica de un diseño y usarla como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. Los casos de prueba derivados del camino básico garantizan que durante la prueba se ejecute por lo menos una vez cada sentencia del programa.

Se realizaron prueba de caja blanca a la funcionalidad crear horario.

Funcionalidad: crear horario

La funcionalidad implementada para crear un horario se muestra en la figura 18.

figura 16: Grafo de Flujo: Funcionalidad crear Horario.



Caminos independientes

Camino 1. (1-2-3-6)

Camino 2. (1-2-4-6)

Camino 3. (1-5-6)

Seguidamente a la construcción del grafo de flujo se procede a efectuar el cálculo de la complejidad ciclomática del código, el cálculo es necesario efectuarlo mediante tres vías o fórmulas que, para concluir que el cálculo fue correcto es necesario que por las tres vías el resultado sea el mismo. A continuación, se muestran las fórmulas aplicadas para calcular la complejidad en cada uno de los métodos.

Tabla 10: Complejidad Ciclomática.

Fórmula 1	Fórmula 2	Fórmula 3
V(G) = (A(Aristas) – N(Nodos)) + 2 V(G)=(7-6)+2=3	V(G) = P (Nodos Predicados) + 1 V(G)= 2+1=3	V(G)=R V(G)=3

Se preparan los casos de prueba que obliguen la ejecución de cada camino del conjunto básico.

Camino 1. (1-2-3-6)

- Entrada: usuario autenticado tiene permiso de escritura en la interfaz crear horario.
- Salida: re-direcciona a la página de inicio del sistema con un mensaje de satisfacción.
- Precondiciones: el usuario debe estar autenticado.

Camino 2. (1-2-4-6)

- Entrada: usuario autenticado tiene permiso de escritura en la interfaz visitante y el visitante existe.
- Salida: interfaz de crear horario.
- Precondiciones: el usuario debe estar autenticado

Camino 3. (1-5-6)

- Entrada: usuario autenticado no tiene permiso de escritura en la interfaz crear horario.
- Salida: interfaz de autenticación del sistema.
- Precondiciones: el usuario debe estar autenticado.

Tabla 11: Casos de prueba de caja blanca.

Caso de prueba para el camino básico # 1		
Nombre de la persona que realiza la prueba: Cléusio de Oliveira Carlos		**Estado de evaluación:** Satisfactorio
Descripción	En este caso no se hace entrada de datos.	
Entradas	contiene ("gestionar horario", $permisos)	
Condición de ejecución	El usuario debe tener permiso de acceso a esta funcionalidad.	
Resultados esperados	Se muestra un listado con todos los horarios existentes en el sistema y las opciones para gestionar.	
Caso de prueba para el camino básico # 2		
Nombre de la persona que realiza la prueba: Cléusio de Oliveira Carlos		**Estado de evaluación:** Satisfactorio
Descripción	En este caso no se hace entrada de datos.	
Entradas	contiene ("gestionar_Horario", $permisos).	
Condición de ejecución	El usuario no tiene permiso de acceso a esta funcionalidad.	
Resultados esperados	Devuelve al usuario a la página principal puesto que no tiene permiso de acceso.	

3.4.2. Pruebas de caja negra

Las pruebas de caja negra, también denominadas como pruebas de comportamiento, se concentran en los requisitos funcionales del software. Son las que se aplican a la interfaz del software. Se utilizan si se conoce la función específica para la que se diseñó el producto, con ella se debe demostrar que cada función es plenamente operacional, mientras se buscan los errores de cada una, teniendo poca relación con la estructura lógica del software (Pressman, 2010).

Específicamente dentro de esta prueba se utilizará la técnica de Partición Equivalente que se basa en una evaluación de las clases de equivalencia para una condición de entrada (Pressman, et al., 2003). En esta técnica se plantea que una clase de equivalencia representa un conjunto de estados validos o inválidos para condiciones de entrada.

Figura 14: Interpretación gráfica de Caja Negra.

A continuación, se muestran las pruebas realizadas a los CUS.

CUS: Gestionar Horarios.

Descripción general:

El caso de uso inicia cuando el administrador de sistema accede al módulo configuración y haz un clic en el menú lateral derecho en la opción horarios, el sistema realiza la operación escogida y permite al administrador crear, eliminar, modificar y listar todos los horarios y finaliza el caso de uso.

Condición de ejecución:

El usuario debe estar autenticado y poseer los permisos.

Escenarios a probar en el CUS:

Tabla 7: Escenario del CUS Gestionar Horarios.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: Crear Horario.	EC 1.1: Crear Horario.	El sistema muestra un formularios con todos los campos que deben ser llenados.	Clic en el botón crear horario " **+"**.
	EC 1.2: Introduce los datos y selecciona la opción aceptar	El usuario debe ingresar los datos del nuevo horario al sistema.	Clic en el botón "Aceptar". Escribir los caracteres deseados en cada campo.
	EC1.3:Campo incompleto.	El sistema muestra un mensaje rellene este campo, y también permite cancelar la acción.	Clic en el botón "Aceptar". Clic en el botón "Cancelar"
SC 2: Mostrar Horario.	EC 2.1: Modificar datos del horario.	El sistema muestra una página con los datos del horario que pueden ser modificados y brinda la posibilidad de modificar los datos y permite también cancelar toda la acción.	Clic en el botón "Editar". Clic en el botón "Aceptar". Clic en el botón "Cancelar".
SC 3: Buscar Horario.	EC 3.1:Buscar Horario.	El sistema busca todos los datos de los horarios que están activo o no activo en el sistema.	Escribe en el filtro "Buscar". Clic en el botón "Si". Clic en el botón "No". Clic en el botón "Todos".

Descripción de las variables:

Tabla 8: Variables CUS Gestionar Horarios.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Buscar	Campo de texto	Si	Pueden ser letras o números.
2	Activo	Botón	No	Debe seleccionar una Opción “SI”, “No”,” Todos”.
3	Anterior	Botón	No	Opción de ir hacia el formulario anterior.

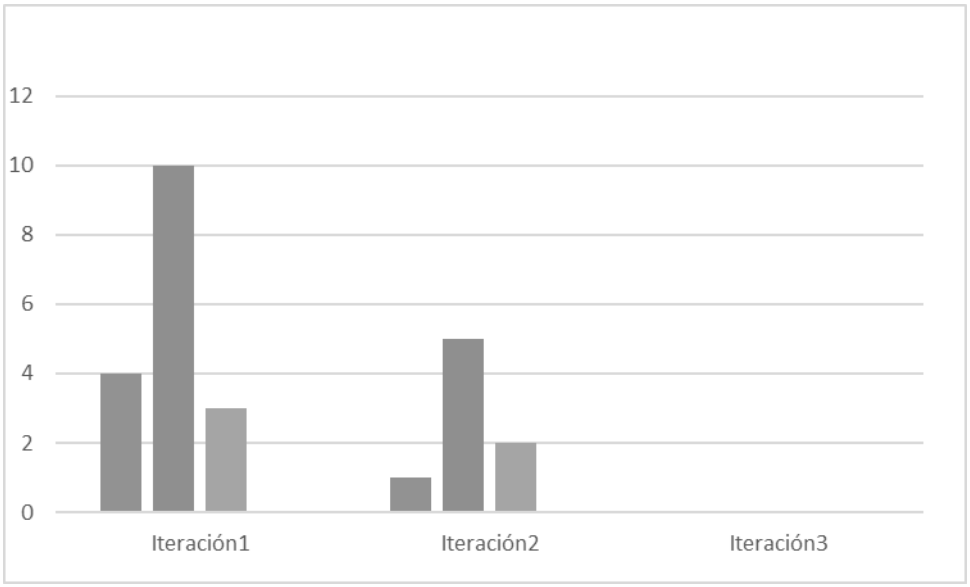
Tabla 9: Matriz de datos del CUS Gestionar Horarios.

ID del Escenario	Escenario	Buscar	Activo	Anterior	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Selecciona la opción Horarios.	V	SI	NA	El sistema muestra un listado con 10 (en caso de que haya menos 10 horarios creados. Mostrando de cada una id, nombre, horario de inicio y fin y se está activo.	Satisfactorio.
EC 1.2	Buscar.	02:05:00	NO	NA	El sistema debe refrescar el listado y mostrar todos horarios que no están activo.	Satisfactorio
EC 1.3	Mostrar todos los horarios.	VA	TODOS	NA	El sistema debe mostrar un listado con todos los horarios que están activo y no.	Satisfactorio.

Resultados de las pruebas

Después de realizar las pruebas al sistema en la primera iteración se encontraron una serie de no conformidades que fueron corregidas, luego se procedió a realizar una segunda iteración, la cual arrojó en algunos casos no conformidades y en otros resultados satisfactorios, luego se realizó una tercera iteración donde todos los resultados fueron satisfactorios, por lo que no se necesitó la realización de una cuarta iteración. Se abarcaron los métodos y técnicas de pruebas expuestas anteriormente en cada una de las iteraciones, arrojando resultados que demuestran la calidad de la aplicación desarrollada. En la primera iteración 15 No Conformidades (NC), de las cuales 10 eran de errores de interfaz, 2 de funcionalidad y 3 de ortografía. Resueltas las no conformidades, se procedió a realizar la segunda iteración detectándose 6 NC, de las cuales 1 era de ortografía y 5 eran errores de interfaz. Al realizarse la tercera iteración no se encontraron NC, dando como resultado que el sistema se encontraba listo para ser desplegado.

figura 15: Resultados de la prueba de iteración.



3.5. Conclusiones Parciales

A lo largo del proceso de implementación y prueba el uso de los estándares de codificación definidos contribuyó a obtener una adecuada uniformidad y legibilidad en el código fuente. El diagrama de componentes dio la posibilidad de ver las dependencias entre los componentes del sistema, además reflejó de manera más detallada la ubicación física de los componentes de acuerdo a la arquitectura empleada. Los métodos de pruebas ejecutados validaron el correcto funcionamiento de la aplicación, cumpliendo con los requisitos especificados y evidenciaron que el software está listo para ser desplegado.

Conclusiones generales

La investigación desarrollada y los resultados obtenidos permiten arribar a las siguientes conclusiones:

- A partir del análisis y diseño del sistema web se obtuvieron los artefactos necesarios para guiar el proceso de desarrollo del mismo.
- El uso del lenguaje de modelado UML facilitó la modelación de los artefactos de los flujos de trabajo de la metodología utilizada.
- El análisis referente a los modelos de control de acceso más utilizados a nivel mundial, arrojó que el control de acceso Basado en Roles es el más adecuado para ser empleado como base de la solución propuesta, a partir de la flexibilidad y neutralidad que posee.
- La obtención de los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos y los artefactos generados, constituyeron elementos claves en la construcción de la propuesta de solución.
- Las pruebas realizadas posibilitaron el correcto funcionamiento de la aplicación permitiendo un fácil manejo por parte de los usuarios, ya que cada uno de los requisitos definidos en el transcurso de la investigación fueron desarrollados.

Recomendaciones

Como parte del proceso de mejora continua del desarrollo de software se proponen las siguientes recomendaciones que tributarían a un producto de mejor calidad. Para el desarrollo de futuras versiones del Sistema de control de acceso a entidades deben tenerse en cuenta los siguientes aspectos

- Incorporar funcionalidades que permitan que se haga el control de acceso mediante el uso de huellas dactilares o con reconocimiento facial, visto que el control solo se realiza mediante el uso de código de barra.
- Permitir personalizar las credenciales generadas por cada institución.
- Añadir un conjunto de reportes gráficos que permitan identificar comportamientos y patrones que puedan ser utilices a cada institución para la toma de decisiones

Referencia Bibliográfica

1. Ingeniería & Desarrollo. Universidad del Norte. 12: 10-23, 2002Instalación y configuración de Apache,un servidor Web gratis. No 12, Colombia : Universidad del Norte, 2002. **Díaz, José Márquez y Sampedro, Leonardo. (2002).**
2. Praticas de software. 2016. **Andres, Grosso.(2016).**
3. Metodología para el desarrollo e implantación de sistemas de información geográfica. 2013. Ari13. **Arias. (2013).**
4. Arquero sistema corporativo. Arquero sistema corporativo. [En línea] 2013. [Citado el: 11 de noviembro de 2019.] https://studylib.es/doc/6314775/descargar-arquero-sistema-corporativo. (2013).
5. infranetworking. infranetworking. [En línea] 4 de enero de 2019. [Citado el: 3 de deciembre de 2019.] https://blog.infranetworking.com/servidor-web/. **Borges, Santiago. (2019).**
6. Cuadernos de Seguridad. Cuadernos de Seguridad. [En línea] 30 de ouctubro de 2019. [Citado el: 30 de novembro de 2019.] https://cuadernosdeseguridad.com/2019/01/como-debe-ser-un-control-de-accesos-infalible/. Nº 99. **Diez, Alfonso. (2019).**
7. historia y evolucion de los lenguajes de programacion**.** Spain : s.n., 2019. **Aranda, Vicente Trigo. (2019).**
8. Historia y evolucion de los lenguajes de programacion**.** 34, spain : s.n., 2018. 1888-6051. **Aranda, vicente Trigo. (2018).**
9. infranetworking**.** Spain : s.n., 2019. **Borges, Santiago. (2019).**
10. Instalación y configuración de Apache. 12, Colombia : s.n., 2002. 0122-3461. **Díaz, José Márquez. (2002).**
11. Access control for rural . Chine Universities. 2013, JYa13. **J, Yang-Feng y Si-Yue Z. (2013).**
12. Sistema de Control y Asistencia a Actividades de la Producción. HAVANA : UCI, 2016. jim16. **Jiménez, Aliú Cuesta. (2016).**
13. UML y Patrones. 2009. **Larman, Craig.(2009).**
14. UML y Patrones.Una introducción al análisis y el diseño orientado a objetos y al proceso unificado. Mexico : s.n., 2015. 053681. (2015).
15. Los gestores de bases de datos más usados en la actualidad. **Marin, Rafael. 2019.**

16. revista digital. revista digital. [En línea] 16 de abril de 2019. 1. **Marín, Rafael. (2019).**
17. Modelado de casos de uso. 2011. **Mediavilla, Elena. (2011).**
18. Information Security Fundamentals. EUA : s.n., 2014. Tho14. **Peltier, Thomas R. (2014).**
19. Information Security Fundamentals. EUA : s.n., 2014. Tho141. (2014).
20. Information Security Fundamentals. EUA : s.n., 2014. Tho142. (2014).
21. GESTIÓN DE LA PREVENCIÓN CONTROL DE ACCESOS. Murcia : s.n., 2016. ARA16. PEREZ, **ARANTXA MORA.(2016).**
22. Security in computing. EUA : s.n., 2010. 978-0-13-2390077-4**. pfleeger, Hidalgo, pozo y P, charles. (2010).**
23. Ingeniería de Software. Un enfoque práctico. Madrid y Carchelejo : McGraw-Hill, 2003. ISBN: 8448132149**. Pressman y S, Roger. (2003).**
24. ingenieria de software un enfoque prático. Mexico : María Teresa Zapata Terrazas, 2010. 978-607-15-0314-5. **Pressman, Roger S. (2010).**
25. Sistema para el control de acceso a los laboratorios del Centro de Tecnologías para la Formación . Havana : UCI, 2015. Ram15. **Ramírez, Yasmani Freixas y Pérez, Virgilio Noa. (2015).**
26. **.** Sistema de Control de Laboratorios . Havana : UCi. UCI**.** Rodríguez**, Luis Olfrides Pérez y Martínez, Daliana González.(2016).**
27. Sistema de Control de Laboratorios. Havana : UCI, 2008. Rod. **Rodríguez, Luis Olfrides Pérez y Rodriguez, Daliana Peréz. (2008).**
28. Metodología de desarrollo para la Actividad productiva de la UCI. Havana : uci, 2015. **Sánchez, Tamara Rodríguez. (2015).**
29. Seguridad informatica. Havana : s.n., 2010 **.Aguirre, jorge Ramio. (2010).**
30. Apriendendo UML en 24 horas. EUA : s.n., 2014. 9789684444638**. shumller, joseph. (2014).**
31. sistema de identificacion por radio frecuencia, codigo de barra y su relacion con la gestion de la cadena de suministro . 116, Bolgota : s.n., 2010, Vol. 26. 65-223. **Espinal, Alexander Correa, Esteban Alvarez , Carlos Lopez y Montoya, Rodrigo Andres gomez. (2010).**
32. Sistema por identificacion por radio frecuencia. 116, Bogota : s.n., 2010, Vol. 26. 65223**. Espinal, Alexander correa. (2010).**
33. Software Engineering. Spain : MIguel-Martin-Romo, 2007. ISBN-84-7829-074-5**. SOMMERVILLE, IAN.(2007).**

34. Technology and Culture**.** Kranzberg, Melvin. 1986**.** No. 3, Georgia : s.n., 1986, Vol. Vol. 27. Mel86.
35. An access control model for cloud computing. Journal of Information Security and Application . 2014, YAT14. YA, **Tounis. (2014).**
36. Guadalupe.LAS TIC EN TRABAJO SOCIAL. [En línea] 19 de marzo de 2012http://ticyts.blogspot.com/2012/03/evolucion-de-lastic-oportunidades-y.html. . [Citado el: 05 de febrero de 2015.]
37. Arquero Sistema corporativo. Software de control y seguridad. Control de acceso y control de horario. [En línea] 15 de enero de 2013.
38. El Desarrollo del Framework Orientado al Objeto. Markiewicz, Carlo, Marcus, Eduardo y Lucena, J.P. (2011).
39. Metodologías Ágiles en el Desarrollo de Software. Recuperado de: http://ima.udg.edu/Docencia/07-08/3105200728/TodoAgil.pdf**. Canós, J., Letelier, P. y Panadés, M. (2006).**
40. https://symfony.es/noticias/2017/11/30/se-publica-symfony-40/revista digital INESEM https://revistadigital.inesem.es/informatica-y-tics/symfony-el-framework-de-moda/Víctor **Manuel Acosta 1/04/2019. Revista 30 de noviembre de 2017 .**
41. .Access control for rural medical and health collaborative working platform [versión electrónica]. The Journal of Chine Universitiesof Posts and Telecommunications 20 (Suppl. 2): 7 – 10. **Yang-Feng J., Si-Yue Z., Zhen H., Mu-Qing L., Ling Y., Jing-Ping N. (2013).**
42. https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados. https://www.um.es/geograf/sigmur/sigpdf/temario. **Rafael Marin (2017).**
43. Sams Teach Yourself PHP, mysql and Apache All in One: STY PHP, mysql Apache AIO_p5. **Sams Publishing. Meloni, J. C. (2012).**
44. AMAG TECHNOLOGY. AMAG TECHNOLOGY. [En línea] Javelin form AMAG technology. http://www.securityinfowatch.com/product/10327856/amag-technologyjavelin.
45. Sistemas de control de acceso y software de control de acceso HID**.** [En línea] SmartCardSystems S.A, 2013 de enero de 2013. http://www.scssa.com.ar/control-deacceso.htm.

Anexos

Figura 17: Diagrama de secuencia del CU autenticar usuario.

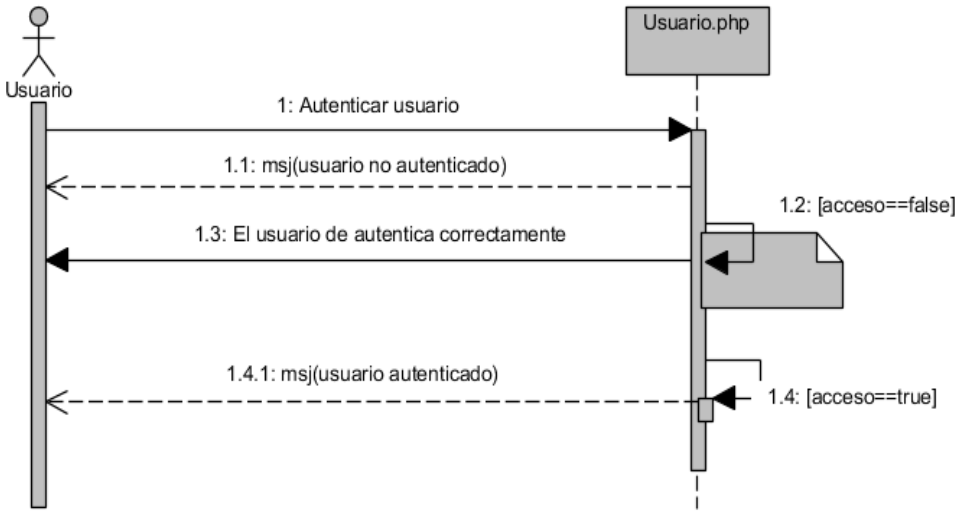
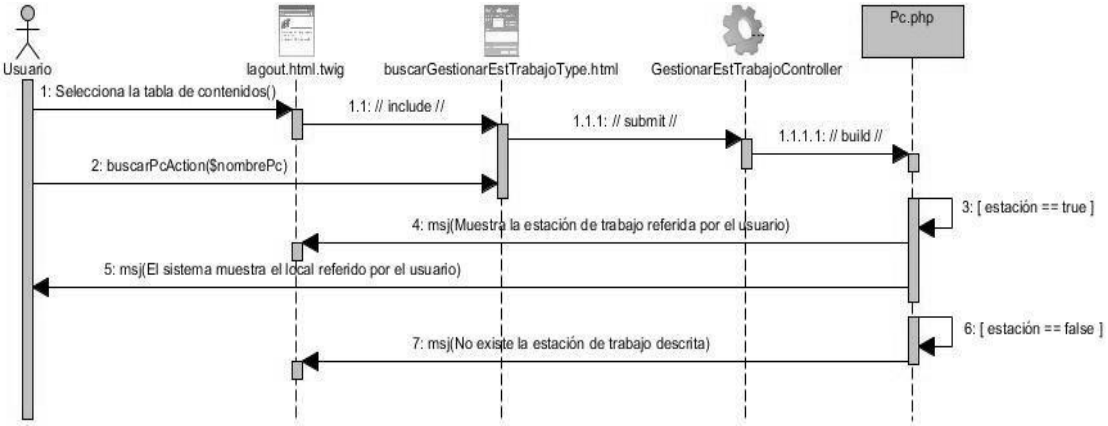


Figura 18: Diagrama de Secuencia buscar Institución.



Anexos 2

Figura 19: Funcionalidad para crear un horario.

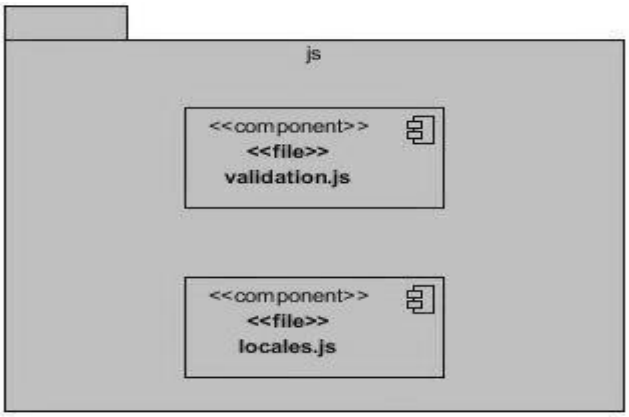
```
public function new(Request $request): Response
{
     1- $horario = new Horario();
     2-$form = $this->createForm(HorarioType::class, $horario);
     $form->handleRequest($request);

    3-if ($form->isSubmitted() && $form->isValid()) {
          4-$entityManager = $this->getDoctrine()->getManager();
          $entityManager->persist($horario);
          $entityManager->flush();

          return $this->redirectToRoute('horario_index');
     }

     5-Return $this->render('horario/new.html.twig', [
          'horario' => $horario,
          'form' => $form->createView(),
     ]);
6}
```

Figura 20: Diagrama de Componentes js.



Anexos 3: Prototipo de IU

Figura 21:Listas de Causas

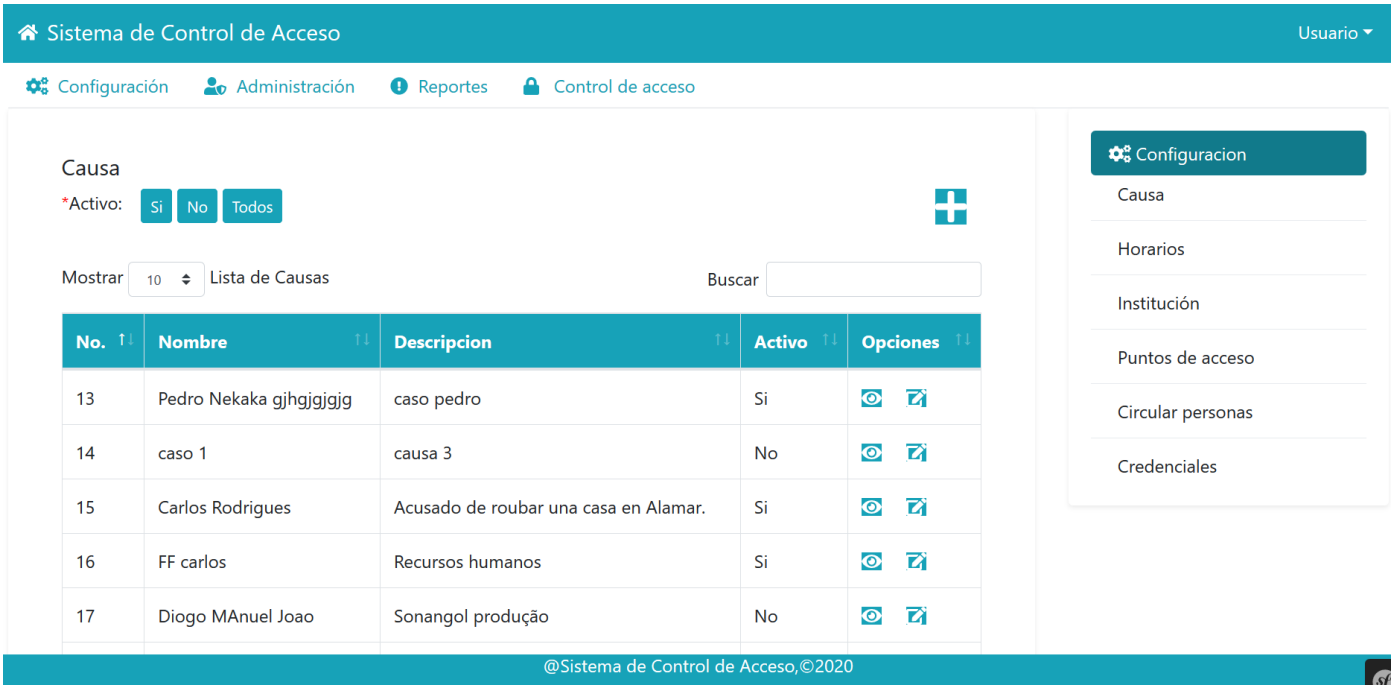


Figura 22: Listas de Instituciones

Anexos 4: Prototipo de IU

Figura 23: Listas de personas circuladas

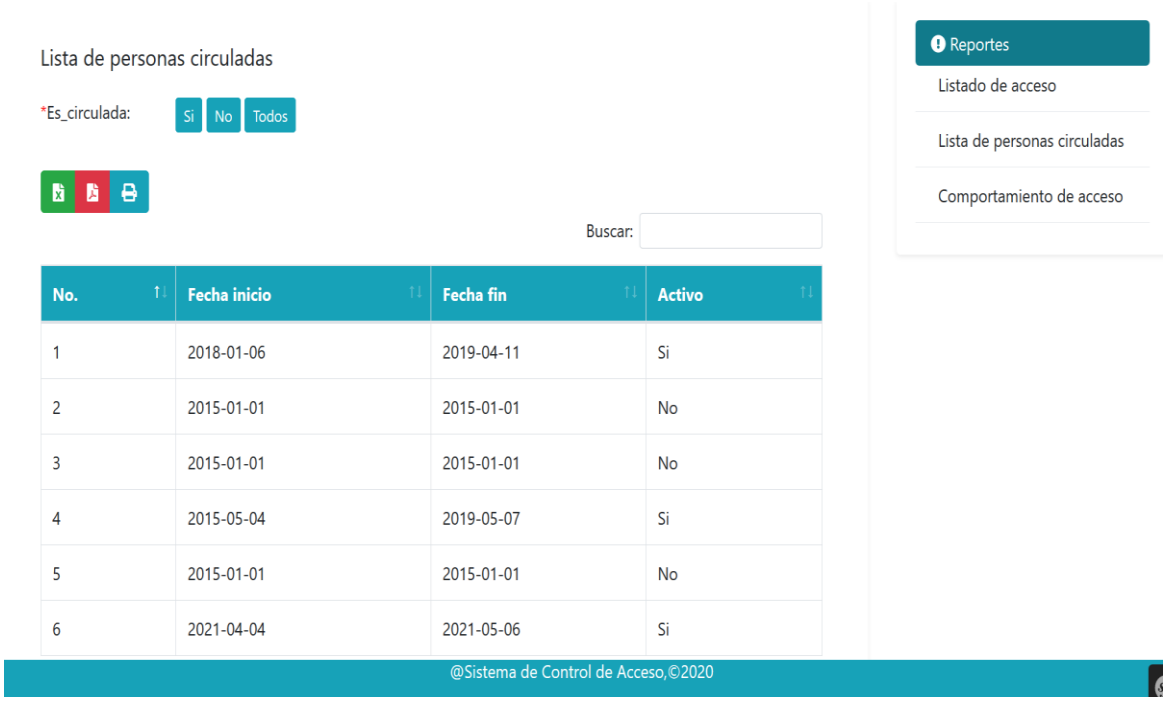


Figura 24: Listas de puntos de acceso