Facultad de Ingeniería
Universidad Nacional de Mar del Plata

Sistema de control de acceso con utilización de tecnologías IoT

Proyecto final de graduación
Ingeniería Informática

Autores: Colautti Bruno y Prieto Cassano Gabriel

Director: Evans Felipe

Codirector: Finochietto Mariano

Fecha: Agosto 2020

Índice

1. Resumen

El presente informe de proyecto final fue desarrollado por los alumnos Bruno Colautti y Gabriel Prieto Cassano de la carrera de Ingeniería Informática de la Universidad Nacional de Mar del Plata. El objetivo del proyecto es realizar un sistema para el control de acceso en el Consorcio Portuario Regional de la ciudad de Mar del Plata. En este documento se encontrará detallada la solución, incluyendo su relevamiento de requerimientos, el análisis, diseño, implementación e instalación de la misma.

El desarrollo se basa en una arquitectura distribuida con uso de tecnologías de IoT, empleando elementos del patrón publish/subscribe, y, ademas, almacenamiento tradicional en base de datos relacional. También se diseñó una interfaz de comunicación con hardware específico, desarrollados por otro grupo de investigación de la Facultad de Ingeniería, para controlar los distintos puestos de ingreso del puerto de la ciudad.

Las herramientas de desarrollo utilizadas son de licencia totalmente libre y se encuentran en repositorios abiertos a la comunidad. La implementación del mismo se encuentra operativa en la actualidad.

Así mismo, se presenta en el Anexo II un paper en el que los alumnos han participado como autores, entre otros, para el VI Congreso Nacional de Ingeniería Informática - Sistemas de Información, CONAIISI 2019.

2. Introducción

La administración de la seguridad en zonas portuarias es un problema de difícil gestión a nivel mundial. La misma debe permitir que un conjunto importante de personas, vehículos y máquinas accedan diariamente a zonas de trabajo de forma ágil, controlando potenciales actividades delictivas.

Según la Comisión Económica para América Latina y el Caribe (CEPAL) "la protección portuaria es un componente esencial de la vulnerabilidad económica del sistema de transporte marítimo de la Región de las Américas, y de la competitividad internacional. La misma debe contribuir a los programas generales de lucha contra el crimen tendientes a combatir el terrorismo y otras amenazas, como el tráfico ilícito de drogas, armas, personas y otras formas de crimen organizado, así como otros ilícitos que afecten la seguridad de la carga y el tráfico marítimo (robos, polizones, contrabando, entre otros), constituyendo una amenaza de explotación ilegítima de los puertos" (Sanchez J.R., Garcia Bernal R, Manosalva Osorio, Rezende Sydney y Sgut M., 2004).

Anteriormente, el Consorcio Portuario Regional de Mar del Plata (de ahora en más CPRMDP) poseía un sistema de control de acceso, el cual no era utilizado debido a que no se ajustaba a sus necesidades. Por lo tanto, no había un registro detallado del ingreso de personas y vehículos, dejando al mismo sin una certificación de la Protección de los Buques y de las Instalaciones Portuarias (PBIP) y, por consiguiente, perdiendo el estado de puerto seguro.

El puerto seguro, plantea el abogado, árbitro y profesor de Derecho Marítimo, José Antonio Pejovés, "es aquel que garantiza que el arribo, estancia y zarpe del buque se dé en condiciones que garanticen que éste soportará los menores riesgos posibles respecto a su seguridad, sin estar expuesto en situaciones normales a peligros que podrían sortearse con maniobras y acciones de navegación idóneas en la zona portuaria" (Pejovés J.A., 2019).

Frente a esta situación, el CPRMDP desea adquirir una solución informática propia que se adapte a sus necesidades, ya que las soluciones predominantes del

mercado son ofrecidas mediante alquiler, dando poca flexibilidad frente a las problemáticas cambiantes de la legislación en seguridad portuaria nacional e internacional.

Al día de hoy, el puerto cuenta con múltiples muelles, cada uno de ellos con su terminal de acceso, distribuidos dentro de la extensa superficie administrada por el CPRMDP. En cada una de estas terminales, las distintas personas deben identificarse con una tarjeta de proximidad propia para determinar si se encuentran habilitadas a ingresar a las zonas restringidas. Estas terminales son controladas por empleados del consorcio, conocidos como guardias. Uno de los objetivos es facilitar la tarea de estos empleados, dándoles una mayor fiabilidad sobre las decisiones referidas al acceso que ellos toman y evitar que se generen demoras y/o inconvenientes en las zonas de ingreso.

3. Marco teórico

Un sistema distribuido puede ser definido como "una colección de elementos de computación independientes que son presentados ante sus usuarios como un único sistema coherente" (Van Steen M. y Tanenbaum A.S., 2017). De esta definición se desprende que los distintos elementos, también llamados nodos, que componen un sistema distribuido, necesitan colaborar para que los usuarios lo perciban como un todo.

Los nodos dentro del sistema pueden ser procesos de software o elementos de hardware, que pueden variar desde computadoras de alto rendimiento a simples sensores. Los mismos deben reaccionar a mensajes externos, procesarlos y producir una respuesta que será enviada a uno o más nodos.

Una de las principales consideraciones, a la hora de tratar con sistemas distribuidos, es la imposibilidad de asumir la presencia de un reloj global. La falta de una referencia común del tiempo lleva a la necesidad de definir distintas estrategias para lograr la coordinación y sincronización de los distintos nodos del sistema.

Otra consideración fundamental es la necesidad de autenticar los distintos nodos que formarán parte del sistema. En rasgos generales, los nodos pueden dividirse en dos grupos: grupos abiertos, donde cualquier nodo puede unirse y comunicarse con los demás, y grupos cerrados, donde es necesario brindar credenciales para poder unirse y comunicarse con el resto de nodos.

En base a estas consideraciones, comúnmente un nodo de un sistema distribuido consiste en un proceso de software provisto con una lista de otros nodos con los que puede comunicarse directamente. A esta forma de organización se la conoce como *overlay network*, y pueden dividirse en dos tipos:

- *Overlay estructurada*, donde cada nodo tiene un conjunto bien definido de vecinos con quienes se puede comunicar. Por ejemplo, los nodos pueden estar organizados en una estructura de árbol o un anillo lógico.

- *Overlay desestructurada*, donde cada nodo tiene un número arbitrario de referencias a otros nodos seleccionados aleatoriamente.

En cualquiera caso, una *overlay network* debe estar, en principio, siempre conectada mediante una red de computadoras. Esto quiere decir que entre dos nodos cualesquiera debe existir siempre un camino de comunicación permitiendo a estos nodos enviarse mensajes entre sí.

A la hora de hablar de un sistema coherente, algunos autores opinan que un sistema distribuido debe presentarse como un sistema único, logrando que los usuarios no noten que en realidad los procesos, la información y el control se encuentran dispersos a través de una red de computadoras. En la práctica esto puede resultar muy complejo, por lo que en ciertas ocasiones se opta por una definición más flexible, en la que se dice que un sistema distribuido tiene que *aparentar* ser coherente. Esto quiere decir que los distintos nodos deben operar como un todo, sin importar donde, cuando y como la interacción entre un usuario y el sistema ocurre.

Para poder mantener esta coherencia a lo largo del tiempo, un sistema debe ser escalable. Con respecto a esto, pueden hacerse tres divisiones en base a:

1. Tamaño: el sistema debe permitir agregar de manera sencilla nuevos usuarios y recursos sin tener una pérdida de performance notable.
2. Geografía: los usuarios y recursos pueden estar separados físicamente por grandes distancias pero el delay en la comunicación no debe ser percibida.
3. Administración: el sistema debe permitir una administración sencilla por más que distintas organizaciones independientes intervengan en el mismo.

Arquitecturas centralizadas Cliente-Servidor

En esta clase de sistemas, los nodos son divididos en dos grupos: un servidor, encargado de brindar un servicio específico, y un cliente, quien solicita servicios a un servidor enviando una solicitud y esperando la correspondiente respuesta, como se esquematiza en la *Figura 3.3*.

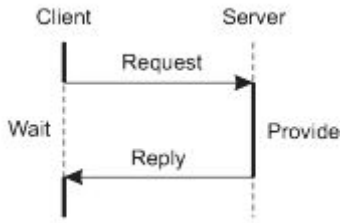


Figura 3.1 - Arquitectura cliente-servidor

Para lograr asegurarse que la respuesta llegue, virtualmente todos los sistemas basados en esta arquitectura utilizan, para su comunicación, el protocolo TCP/IP. Esto es así ya que, a diferencia del protocolo UDP, cada vez que el cliente requiere un servicio, primero se establece una conexión con el servidor para luego enviar la solicitud sobre un canal confiable. Generalmente el servidor utiliza la misma conexión para enviar su respuesta, la cual luego es cerrada.

A la hora de diseñar un servidor, un aspecto clave es definir los puntos en donde los clientes se conectarán; estos puntos son conocidos como **puertos**. Cada proceso servidor que se encuentre corriendo en un equipo brindará su servicio a través de un puerto específico, distinto al de los demás servidores. Para que los clientes conozcan en qué puerto se brinda cada servicio se pueden optar por alguno de los siguientes enfoques:

- Asignar cada servicio al puerto estándar definido por la Internet Assigned Numbers Authority (IANA). Esto se conoce como “servicios bien conocidos”.
- Ejecutar un proceso conocido como *daemon* en los equipos donde los servidores corren, el cual es el encargado de anunciar el puerto sobre el cual cada servicio está siendo brindado. Un cliente primero debe contactarse con el daemon y solicitar el puerto para el servicio que desea consumir.

Otro aspecto importante a tener en cuenta al momento del diseño es el manejo de los estados por parte del servidor. Un servidor **sin estado** no mantiene información del estado de sus clientes, y puede cambiar su propio estado sin necesidad de informar a ninguno de ellos. Este tipo de servidores responde a la solicitud entrante y, una vez la misma es finalizada, olvida todo lo relacionado al cliente que la realizó.

Un servidor **con estado** persiste la información de sus clientes; esto quiere decir que la información debe ser explícitamente eliminada por el servidor. Si el servidor sufre un error que implique la necesidad de un reinicio, deberán recuperarse los estados más actualizados de los clientes para poder garantizar el correcto funcionamiento a la hora de responder futuras solicitudes.

Arquitecturas descentralizadas Peer-to-peer

Desde una perspectiva de alto nivel, los nodos que componen un sistema con esta arquitectura son todos iguales. Esto significa que las funcionalidades que se necesitan realizar están distribuidas a lo largo de todo el sistema, y no se encuentran centralizadas en partes del mismo. Como consecuencia, mucha de las interacciones entre procesos será simétrica: cada proceso actuará como cliente y servidor al mismo tiempo, situación a la que se refiere como sirviente.

A la hora de organizar los nodos y las conexiones entre los mismos pueden adoptarse distintos enfoques, dependiendo de los requerimientos del sistema. Por ejemplo, se puede optar por una arquitectura donde todos los nodos sean pares, como en la *Figura 3.2*, o también, algunos nodos pueden tener mayor jerarquía, como en la *Figura 3.3*.

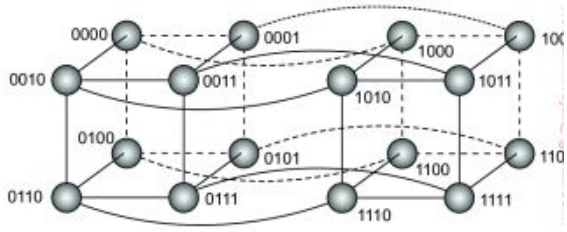


Figura 3.2 - Arquitectura Peer-to-peer regular

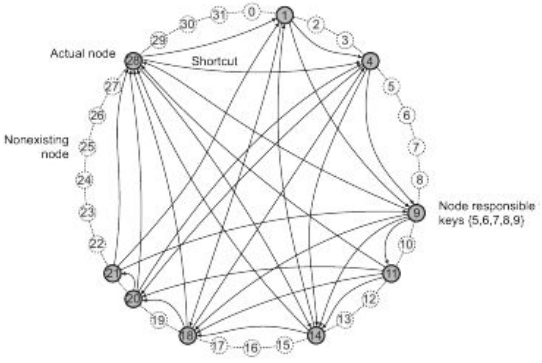


Figura 3.3 - Arquitectura Peer-to-peer con jerarquía

Arquitecturas de publicación-suscripción

La arquitectura de publicación-suscripción es un estilo de aplicación de mensajería en la que los proveedores de información (publicadores) no tienen enlace directo a los consumidores específicos de esa información (suscriptores), sino que las interacciones entre publicadores y suscriptores las controla una capa intermedia (middleware) de publicación/suscripción, como se muestra en la *Figura 3.4*.

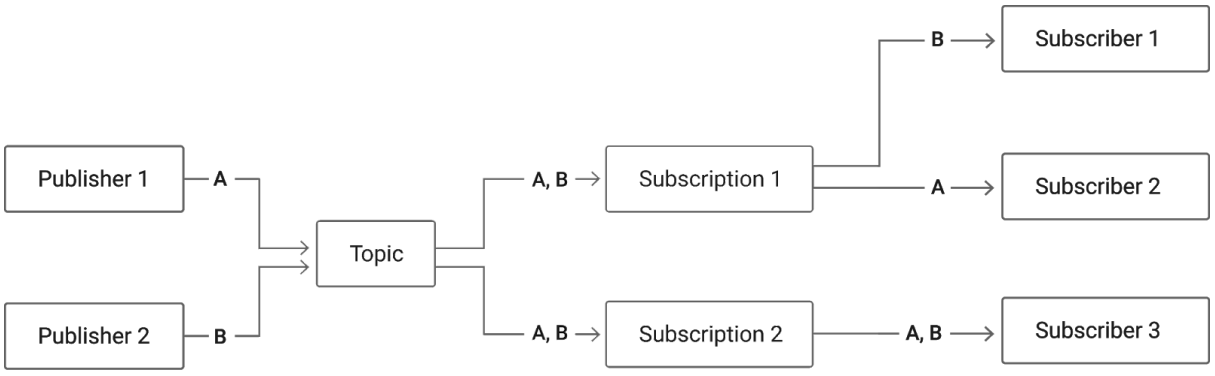


Figura 3.4 - Arquitectura publicación-suscripción

En un sistema de publicación/suscripción, un publicador no necesita saber quién utiliza la información (publicación) que proporciona, y un suscriptor no necesita saber quién proporciona la información que recibe como resultado de una suscripción. Las publicaciones se envían de los publicadores al middleware, las suscripciones se envían de los suscriptores al middleware, y es este último quien envía las publicaciones a los suscriptores. Esto permite comunicar componentes de distinta naturaleza de una manera simple y confiable.

Internet de las Cosas

El Internet de las Cosas, abreviado IoT (del inglés *Internet of Things*), hace referencia a una red de objetos físicos que llevan sensores integrados, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de internet. Dichos dispositivos pueden ser desde objetos domésticos cotidianos, como un sistema de luces, hasta sofisticadas herramientas industriales.

Aunque la idea de IoT existe desde hace mucho tiempo, una serie de avances recientes en diversas tecnologías la ha hecho realidad.

- **El acceso a tecnología de sensores de bajo coste y baja potencia.** Los sensores asequibles y fiables hacen que la tecnología de IoT sea posible para más fabricantes.
- **Conectividad.** Un conjunto de protocolos de red para Internet ha hecho que sea fácil conectar sensores a la nube.
- **Plataformas de Cloud Computing.** El aumento de la disponibilidad de las plataformas en la nube permite que los usuarios accedan a la infraestructura que necesitan para ampliar la capacidad sin tener que gestionarlo todo.
- **Machine learning y analítica.** Con los avances logrados en machine learning y en analítica, junto con el acceso a enormes cantidades de datos de una gran variedad almacenados en la nube, las empresas pueden reunir información más rápido y de forma más sencilla.
- **Inteligencia artificial (IA) conversacional.** Los avances en redes neuronales han llevado el procesamiento de las lenguas naturales (NLP) a los dispositivos de IoT (por ejemplo, los asistentes personales Alexa, Cortana y Siri) y los han convertido en dispositivos atractivos, asequibles y viables para el uso doméstico.

4. Análisis FODA

Al inicio del proyecto se realizó un análisis de los factores internos y externos que tienen influencia sobre el mismo. Por ello, se analizaron las fortalezas, las oportunidades, las debilidades y las amenazas. Así, se pueden potenciar las fortalezas y buscar alternativas a las debilidades, como aprovechar las oportunidades externas y prevenir las amenazas que puedan surgir.

Fortalezas

- Debido a experiencia en trabajos previos, se cuenta con una buena comunicación entre los integrantes del equipo de trabajo. Esto ayuda a la hora de tomar decisiones y evita conflictos internos que puedan atentar contra el proyecto.
- Al no haber ninguna limitación impuesta por el cliente con las tecnologías a utilizar, se seleccionan aquellas con las que ya se tenga experiencia y, por lo cual, hayan mayores conocimientos sobre las mismas.
- Las tecnologías que han sido aplicadas tienen en sus haberes muchos casos de éxito. Además, son tecnologías conocidas en el ámbito y que cuentan con soporte.
- Se tiene un contacto directo con el solicitante del sistema, con acceso a muchas reuniones para relevar información. Así mismo, los usuarios finales forman parte del relevo, brindando una retroalimentación proactiva durante el desarrollo del proyecto.
- El director del proyecto cuenta con conocimientos y experiencia en todos los temas referidos al desarrollo. Esto implica una alta participación en ambos grupos de desarrollo, tanto el de programación del sistema como el de electrónica, brindando capacidad de resolución y soporte a todos los integrantes.

Oportunidades

- Es un ingreso en un mercado y una forma de comercialización nueva. Al ser una solución hecha a medida para el consorcio portuario, se puede llegar a replicar la solución en otros consorcios de distintos puntos del país.
- Utilización de tecnologías de IoT en el contexto de la seguridad portuaria, expandiendo así sus campos de aplicación. Si el proyecto tiene éxito, puede generar un impacto notable dentro del sector.
- Al trabajar en conjunto con el grupo de electrónica, hay una oportunidad de aprendizaje y enriquecimiento en un campo desconocido, potenciando la capacidad profesional.

Debilidades

- Es la primera oportunidad profesional de los integrantes del equipo de programación para realizar un proyecto con un cliente real.
- No se cuenta con una estandarización de los procesos, por lo cual los mismos deberán ser definidos entre las partes previo al desarrollo del sistema.
- Hay un gran número de usuarios que van a ejercer sus actividades con el sistema y, por ello, brindar requisitos. Esto puede ser contraproducente al momento de determinar los requerimientos del sistema, ya que las diferentes opiniones de los usuarios sobre el flujo de trabajo puede generar incongruencias.
- No se cuenta con experiencia previa en desarrollo de hardware especializado, ni integración de los mismos a un sistema de este calibre.
- Ya que el proyecto está integrado por estudiantes de distintas carreras y/o distintos ámbitos, existe la posibilidad de contar con dificultades a la hora de fijar reuniones donde participen todos los integrantes debido a incompatibilidades horarias.

Amenazas

- Al contar con experiencia en un sistema anterior, los usuarios pueden mostrarse reacios a adoptar la nueva solución.
- Siendo el cliente un ente público, existe una gran burocracia a la hora de realizar compras y/o pagos, por lo que las compras del hardware necesario podrían verse retrasadas, haciendo que el proyecto se paralice en alguna de sus etapas.
- El sistema financiero del país es inestable, lo que podría provocar que la solución, viable en un comienzo, deje de serlo ante un cambio brusco en las condiciones económicas argentinas.
- Las actividades realizadas dentro del puerto se rigen bajo distintas legislaciones portuarias (ya sean nacionales, provinciales y/o municipales). Cualquier potencial cambio en esta legislación podría traer la necesidad de realizar modificaciones en el sistema para adaptarse a ellas.
- Al abarcar un gran número de procesos de distinta naturaleza, la curva de aprendizaje del sistema para un usuario inexperto puede resultar elevada.

5. Especificaciones a resolver

Como se mencionó previamente, el CPRMDP se encontraba en la posición de perder la certificación de PBIP, colocando al puerto como no seguro. Para recuperar el estado puerto seguro, se solicitó un sistema que sea capaz de determinar si una persona que intenta acceder a las zonas seguras del puerto está habilitada a hacerlo; así como establecer, en caso de ser necesario, si su vehículo posee las habilitaciones necesarias para circular dentro de dichas zonas. De igual forma, se debe llevar un registro de toda la información necesaria sobre el ingreso a fin de ser auditable.

Para el acceso se requiere la utilización de un conjunto ya existente de lectores para tarjetas por aproximación. Además de la reutilización de dichos sensores, un requerimiento del cliente fue la necesidad adaptar los mismos para que puedan operar offline por un tiempo prudencial ante un fallo o indisponibilidad eventual de la red local. También solicitó la posibilidad de cambiar de proveedor de hardware sin que esto implique realizar modificaciones en el sistema.

Respecto al alcance, el proyecto contempla todo lo relacionado al control de acceso a las zonas seguras del puerto, así como auditorías sobre los mismos. No está contemplada la búsqueda de anomalías en cuanto a inconsistencias en el histórico de entradas (por ejemplo, dos entradas sucesivas de una misma persona sin una salida entre ellas), la asociación de los trabajadores con los barcos en los cuales los mismos trabajan y el seguimiento de la persona cuando se encuentra dentro de una zona restringida.

Especificaciones funcionales

El sistema será afectado por las acciones que realizan los diferentes tipos de usuarios, estos son: los administradores, los guardias, los trabajadores que ingresan los sectores protegidos, los operadores de monitoreo y la gerencia. A continuación se describirán los casos de uso para los usuarios mencionados.

Administración

Los operadores del área de administración son los encargados de dar las altas, bajas y realizar las modificaciones a las diferentes entidades del sistema, de acuerdo a la información y los requisitos brindados por el representante de la empresa. Esta información será la que permita determinar a los guardias si un trabajador tiene permitido o no el ingreso a las zonas restringidas del puerto.

Para satisfacer estas necesidades, se definen los siguientes casos de uso:

Nombre	**ABM de trabajador regular**
Actores	Usuario administrativo, trabajador
Tipo	Primario
Descripción	Registrar la información de un trabajador que tendrá un ingreso regular a una o más zonas restringidas del puerto. El trabajador debe ser único en el sistema, utilizando para su identificación el número de CUIL si el trabajador es argentino o su pasaporte si es extranjero. Suspender al trabajador por incumplimientos, documentación vencida o documentación faltante.

Curso normal de los eventos para un alta de persona	
Acción del actor	**Respuesta del sistema**
1. El trabajador se presenta en la oficina de administración con la planilla de la información personal y la documentación requerida.	
2. El usuario administrativo corrobora que la información sea correcta, luego inicia proceso de alta de trabajador.	3. Muestra formulario de creación de trabajador.
4. El usuario administrativo ingresa la información necesaria y toma una foto del trabajador. Además le asigna un número de PIN.	5. Valida los datos y los guarda.
7. El usuario administrativo solicita impresión de una credencial de ingreso con la foto, nombre, nivel de riesgo y PIN del trabajador.	8. Imprime credencial.

9. El usuario administrativo entrega la nueva credencial al trabajador; luego, el trabajador se retira.	

Cursos alternos

- Paso 2. La información y/o documentación presenta algún tipo de error, por lo que el usuario administrativo informa al trabajador las correcciones necesarias. El trabajador se retira y el proceso de alta no inicia.
- Paso 5. El sistema encuentra uno o más errores en los datos ingresados, por lo que notifica al usuario administrativo acerca de los mismos. El usuario administrativo corrige la información.

Nombre	**ABM de una empresa**
Actores	Usuario administrativo, representante de la empresa
Tipo	Primario
Descripción	Registrar la información de una empresa cuyos empleados tendrán acceso a una o más zonas restringidas del puerto. La empresa debe ser única en el sistema. Suspender a la empresa por incumplimientos, documentación vencida o documentación faltante.

Nombre	**ABM de un vehículo**
Actores	Usuario administrativo, representante de la empresa
Tipo	Primario
Descripción	Registrar la información de un vehículo en el sistema. El vehículo debe ser único en el sistema y debe pertenecer a una empresa existente. Suspender al vehículo por incumplimientos, documentación vencida o documentación faltante.

Nombre	**Asociación trabajador y vehículo/s**
Actores	Usuario administrativo, trabajador, representante de la empresa
Tipo	Primario
Descripción	Habilitar el ingreso del trabajador utilizando uno o más vehículos, con previa autorización de la empresa responsable de ellos. El trabajador puede o no ser empleado de dicha empresa.

Nombre	**Asociación trabajador y empresa**
Actores	Usuario administrativo, trabajador
Tipo	Primario
Descripción	Habilitar al trabajador para trabajar en las zonas restringidas con el amparo de una empresa determinada. El trabajador no debe estar asociado previamente a la empresa y debe presentar toda la documentación requerida según su tipo de trabajo.

Nombre	**Asignación de PIN a trabajador**
Actores	Usuario administrativo, trabajador
Tipo	Primario
Descripción	Asignar a un trabajador un PIN, el cual va a ser su identificación de ingreso. Asimismo, dar la posibilidad de asociar accesos específicos a ese trabajador.

Nombre	**Baja, robo, rotura o pérdida de credencial**
Actores	Usuario administrativo, trabajador
Tipo	Primario
Descripción	Asignar un nuevo PIN único de ingreso a un trabajador ya asociado

	previamente a una empresa. Se imprime la nueva credencial y se la entrega al trabajador para que pueda acceder a las zonas restringidas. A su vez, es posible dar de baja el PIN que ya no será utilizado por el trabajador, haciendo que el mismo no pueda ser empleado para acceder a través de ninguna de las zonas seguras del puerto.

Nombre	**Verificar documentación**
Actores	Usuario administrativo
Tipo	Secundario
Descripción	Consultar el estado de la documentación presentada para las distintas entidades del sistema, verificando que la misma se encuentre completa y actualizada (sin vencer).

Nombre	**Estado de PIN**
Actores	Usuario administrativo
Tipo	Secundario
Descripción	Verificar el estado de un PIN de un trabajador específico, teniendo en consideración el estado de la empresa, el estado de la documentación del trabajador, el período de habilitación del PIN y el período de habilitación del trabajador en la empresa.

Guardias

Los guardias son los encargados de controlar los distintos accesos a las zonas restringidas, validando que las credenciales de los diferentes trabajadores se encuentren vigentes, al igual que sus vehículos. En ciertas ocasiones, personas invitadas por la gerencia, que no se encuentran registradas como trabajadores, deben ingresar al puerto y dichas visitas deben quedar registradas.

Para satisfacer estas necesidades, se definen los siguientes casos de uso:

Nombre	**Ingreso de un trabajador por un acceso peatonal**
Actores	Guardia, trabajador
Tipo	Primario
Descripción	El trabajador se presenta en uno de los puestos de ingreso con su credencial y el guardia le permite o deniega el acceso en base a la información del sistema.

Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1. El trabajador aproxima su credencial al lector.	2. Muestra en pantalla la información del trabajador y el estado de su PIN.
3. El guardia le informa al trabajador la decisión sobre el ingreso (habilitado o denegado).	
4. El guardia indica en el sistema la decisión tomada.	5. Guarda un registro con la información del acceso.
6. El trabajador acata la decisión del guardia (ingresa al área protegida o se retira).	

Cursos alternos

- Paso 1. El trabajador no tiene la tarjeta física. El guardia lo busca en el sistema por número de PIN o número de DNI.
- Paso 4. El sistema indica que el PIN está habilitado y el guardia no permite el acceso, por lo que debe ingresar la razón de esta decisión; el trabajador se retira. Si el sistema indica que el PIN estaba inhabilitado y el guardia permite el ingreso, debe ingresar la razón de esta decisión; el trabajador ingresa.

Nombre	**Ingreso de un trabajador por un acceso vehicular**
Actores	Guardia, trabajador
Tipo	Primario
Descripción	El trabajador se presenta en uno de los puestos de ingreso con su credencial conduciendo un vehículo, pudiendo concurrir acompañado de más trabajadores. El guardia le permite o deniega el acceso en base a la información del sistema.

Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1. El trabajador aproxima su credencial al lector.	2. Muestra en pantalla la información del trabajador, el estado de su PIN y los vehículos habilitados para el ingreso.
3. El guardia indica el vehículo con el que ingresa el trabajador. Si el trabajador ingresa con acompañantes, el guardia ingresa las credenciales de ellos.	
4. El guardia le informa al trabajador/es la decisión sobre el ingreso (habilitado o denegado).	
5. El guardia indica en el sistema la decisión tomada.	6. Guarda un registro con la información del acceso, con el vehículo utilizado.
7. El/los trabajador/es acata/n la decisión del guardia (ingresa al área protegida o se retira).	
Cursos alternos	
❖ Paso 1. El trabajador no tiene la tarjeta física. El guardia lo busca en el sistema por número de PIN o número de DNI. ❖ Paso 3. El vehículo con el que se intenta ingresar no se encuentra dentro de los vehículos habilitados para el trabajador. El trabajador se retira. ❖ Paso 5. El sistema indica que el PIN está habilitado y el guardia no permite el acceso, por lo que debe ingresar la razón de esta decisión; el trabajador se retira. Si el sistema indica que el PIN estaba inhabilitado y el guardia permite el ingreso, debe ingresar la razón de esta decisión; el trabajador ingresa.	

Nombre	**Ingreso de un trabajador por un acceso de camiones**
Actores	Guardia, trabajador
Tipo	Primario
Descripción	El trabajador se presenta en uno de los puestos de ingreso con su credencial conduciendo un camión, pudiendo concurrir acompañado de más trabajadores. En caso de ingresar un container, debe presentar la información del mismo. El guardia le permite o deniega el acceso en base a la información del sistema.

Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1. El trabajador aproxima su credencial al lector.	2. Muestra en pantalla la información del trabajador, el estado de su PIN y los camiones habilitados para el ingreso.
3. El guardia indica el camión con el que ingresa el trabajador. Si el trabajador ingresa con acompañantes, el guardia ingresa las credenciales de ellos.	
4. Si el camión ingresa con un contenedor, se ingresa la información del mismo.	
5. El guardia le informa al trabajador/es la decisión sobre el ingreso (habilitado o denegado).	
6. El guardia indica en el sistema la decisión tomada.	7. Guarda un registro con la información del acceso, con el camión utilizado y, en caso de tenerlo, el contenedor.
8. El/los trabajador/es acata/n la decisión del guardia (ingresa al área protegida o se retira).	
Cursos alternos	
❖ Paso 1. El trabajador no tiene la tarjeta física. El guardia lo busca en el sistema por número de PIN o número de DNI. ❖ Paso 3. El vehículo con el que se intenta ingresar no se encuentra dentro de los vehículos habilitados para el trabajador. El trabajador se retira. ❖ Paso 4. El container no ha ingresado previamente al puerto, por lo que el mismo es dado de alta en el sistema. ❖ Paso 6. El sistema indica que el PIN está habilitado y el guardia no permite el acceso, por lo que debe ingresar la razón de esta decisión; el trabajador se retira. Si el sistema indica que el PIN estaba inhabilitado y el guardia permite el ingreso, debe ingresar la razón de esta decisión; el trabajador ingresa.	

Nombre	**Ingreso de un invitado por un acceso peatonal**
Actores	Guardia, persona invitada
Tipo	Secundario
Descripción	El invitado se presenta en uno de los puestos de ingreso y el guardia le permite o deniega el acceso en base a la información del sistema.

Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1. El invitado se presenta en el puesto de ingreso con su número de DNI.	2. Muestra en pantalla la información del invitado y el estado de sus invitaciones. Además, muestra el estado de la lectura de la normativa de conducta.
3. El guardia le informa al invitado la decisión sobre el ingreso (habilitado o denegado).	
4. El guardia indica en el sistema la decisión tomada.	5. Guarda un registro con la información del acceso.
Cursos alternos	
❖ Paso 3. La normativa de conducta nunca ha sido leída, el guardia debe leerle la normativa al invitado previo su ingreso. ❖ Paso 4. El sistema indica que el invitado está habilitado y el guardia no permite el acceso, por lo que debe ingresar la razón de esta decisión; el invitado se retira. Si el sistema indica que el invitado estaba inhabilitado y el guardia permite el ingreso, debe ingresar la razón de esta decisión; el invitado ingresa.	

Nombre	**Accesos y egresos previos en la terminal de trabajo**
Actores	Guardia
Tipo	Secundario
Descripción	Corroborar los ingresos y egresos, de las últimas 24 horas, de los trabajadores por el acceso correspondiente al puesto de trabajo donde se encuentra el guardia.

Nombre	**Registro de eventos en el Libro del Guardia**
Actores	Guardia
Tipo	Secundario
Descripción	El guardia registra todo evento de importancia en un histórico., el cual no puede ser editado. Se debe dejar constancia de la fecha y hora, la terminal de trabajo y el nombre del guardia que realizó el registro.

Nombre	**Visualización de cámaras de seguridad**
Actores	Guardia
Tipo	Secundario
Descripción	En cada puesto de trabajo se cuenta con cámaras de seguridad, las cuales deben ser visualizadas en la interfaz del guardia para un mayor panorama de su puesto.

Operadores de monitoreo

Este tipo de usuario está encargado de la seguridad, desde un lugar remoto, de toda la zona del CPRMDP. Por ello, tienen acceso de lectura a los datos ingresados por los usuarios de administración y los guardias.

Para satisfacer estas necesidades, se definen los siguientes casos de uso:

Nombre	**Ver los datos de empresas, trabajadores y vehículos**
Actores	Operador de monitoreo
Tipo	Primario
Descripción	En su puesto de trabajo tiene acceso de solo lectura a los datos cargados por el usuario de administración. Por lo que puede acceder a los listados de las empresas, trabajadores y vehículos que conforman el sistema, como así el resto de sus datos, documentación y vencimientos.

Nombre	**Estado de PIN**
Actores	Operador de monitoreo
Tipo	Primario
Descripción	Verificar el estado de un PIN de un trabajador específico, teniendo en consideración el estado de la empresa, el estado de la documentación del

	trabajador, el período de habilitación del PIN y el período de habilitación del trabajador en la empresa.

Nombre	**Histórico de accesos y egresos**
Actores	Operador de monitoreo
Tipo	Primario
Descripción	Efectuar búsquedas sobre los ingresos y egresos de los trabajadores y vehículos a las zonas restringidas por cualquier terminal de acceso, para así realizar un seguimiento y generar reportes.

Nombre	**Control de libro del guardia**
Actores	Operador de monitoreo
Tipo	Secundario
Descripción	Realizar un seguimiento de las entradas que realizan los guardias sobre los libros que llevan. Tener conocimiento de cuál es el guardia que se encuentra en una determinada terminal.

Gerencia

Los usuarios de la gerencia tienen todas las mismas funcionalidades que tiene un usuario de administración y las de un operador de monitoreo: pueden dar de alta, como modificar datos, de empresas, trabajadores y vehículos; pueden ver los ingresos y egresos de los trabajadores, como también las entradas de los libros de los guardias.

Como funcionalidad extra, estos usuarios pueden cambiar el nivel de riesgo de la zona restringida. Hay tres niveles de riesgo, cada uno de ellos limita la entrada de gente que puede ingresar, según el rango que estos ocupan dentro del consorcio.

Además, estos usuarios son los encargados de dar las altas a personas invitadas que pueden acceder a las zonas restringidas. Las personas invitadas tienen permitido ingresar a las zonas restringidas por un periodo de tiempo establecido, y no tienen un PIN para su ingreso, sino que se presentan al guardia con su DNI y, si están dados de alta y se presentan dentro del periodo de tiempo, pueden pasar.

Para satisfacer estas necesidades, se definen los siguientes casos de uso:

Nombre	**Cambio de nivel de riesgo**
Actores	Usuario de gerencia
Tipo	Primario
Descripción	El usuario cambia el nivel de riesgo del sistema. Los trabajadores que tiene ese nivel de riesgo o uno más restringido pueden acceder. Los trabajadores que poseen un nivel de riesgo más bajo tienen prohibida la entrada.

Nombre	**Alta de personas invitadas**
Actores	Usuario de gerencia, persona invitada
Tipo	Secundario
Descripción	La persona invitada brinda sus datos y autorización. El usuario de gerencia carga los datos de la persona en el sistema y un período en el cual esta persona está habilitada a ingresar.

Especificaciones no funcionales

Como en cualquier software, existen características transversales a todos los módulos del sistema. Al igual que en toda gran organización, los puestos de trabajo varían, tanto en cantidad como en ubicación, por ello se necesita una rápida instalación de nuevas terminales de trabajo para poder empezar a utilizar el sistema, sin la necesidad de una persona externa para tal fin. Complementariamente, el sistema debe

ser multiplataforma, pudiendo acceder al mismo sin depender de un sistema operativo determinado.

Al existir un gran volumen de usuarios que utilizan el sistema y modifican los datos del mismo, es necesario registrar las distintas acciones que cada usuario individual realiza, así como el momento y terminal en que la efectuó. Con esta información, es posible realizar auditorias que permitan resolver conflictos o dudas de los administradores sobre cualquier movimiento realizado sobre el sistema.

Las terminales de los guardias ya cuentan con lectores de credenciales. Debido a esto, se requiere adaptar los mismos para su uso en el nuevo sistema. Así mismo, es necesario permitir la adquisición de nuevos sensores sin dependencia de un único proveedor.

El CPRMDP administra una gran cantidad de metros cuadrados. Las oficinas que tienen acceso al sistema están dispersas en diferentes edificios. Además, las terminales de acceso se encuentran distribuidas por toda el área. Para conectar todos estos equipos existe una red local sobre la cual el sistema deberá trabajar. Sin embargo, al ser necesaria una disponibilidad extremadamente alta, las distintas terminales deben contar con la capacidad de trabajar offline por un tiempo prudencial ante una indisponibilidad eventual de la red local, permitiendo que los accesos y egresos de las zonas restringidas se realicen con normalidad.

6. Modalidad de trabajo

Con la finalidad de realizar un trabajo óptimo, las tareas se separaron en dos grupos, cada uno de los cuales estaba más especializado en la tarea asignada. Por un lado, se tenía al grupo de electrónica encargados del diseño y construcción de los lectores de credenciales. Por otro lado, el equipo de programación, integrado por los autores de este Trabajo Final. Como nexo entre ambos equipos, se contaba con el director del proyecto, con los conocimientos necesarios en ambas disciplinas.

Se realizó una reunión con la gerencia para tener un conocimiento global de lo que se requería para cada módulo y, luego, se procedió a relevar información de los diferentes operarios del mismo. En primer lugar, fue desarrollado el *módulo de administración*, ya que es el encargado de gestionar la información de los trabajadores; luego, se procedió con el *módulo de guardias*, en el cual se cumple la función de determinar quién ingresa a la zona restringida; finalmente, los *módulos de monitoreo y de gerencia*, los cuales son de control y comparten funcionalidad con los módulos anteriores.

Para tener mayor contacto con el cliente, y así generar una mejor retroalimentación sobre el sistema, se decidió utilizar una metodología ágil de desarrollo iterativa e incremental. En cada iteración, fue posible realizar reuniones con el cliente para relevar información sobre alguna funcionalidad a desarrollar previo a realizar la programación de la misma. Una vez desarrollada, en una nueva reunión, era posible validar el correcto avance del proyecto, así como resolver dudas o realizar modificaciones. Finalmente, se iniciaba una nueva iteración con otra funcionalidad a resolver.

A la hora de seleccionar qué funcionalidad se iba a desarrollar, tuvieron prioridad aquellas que en los casos de uso fueron definidas de tipo *Primario*, para luego seleccionar las de tipo *Secundario*, y así concluir con las funcionalidades del módulo.

Una vez un módulo se encontraba finalizado, el mismo era instalado y, luego, se realizaba una capacitación para todos los usuarios del módulo. Gracias al uso de los

distintos usuarios, fue posible detectar errores y cambios respecto a la usabilidad del mismo antes de la finalización del sistema.

Al ser dos las personas involucradas en el desarrollo del software y contar con distinta disponibilidad horaria, fue necesario realizar una gran parte del proyecto de manera remota. Para lograr esto, se utilizaron distintas herramientas para dividir, organizar y sincronizar el trabajo realizado.

La principal herramienta utilizada para versionar el código fue *Git* y, para poder sincronizar a través de internet el repositorio creado, se utilizó *GitHub*. A la hora de escribir el código se optó por utilizar el editor de texto *Visual Studio Code*, el cual ofrece una diversa biblioteca de extensiones para adaptarlo a las necesidades concretas del proyecto. Entre estas extensiones, merece la pena destacar la extensión *Live Share*, la cual permite a dos o más personas editar en tiempo real los archivos alojados en una única computadora a través de la red.

7. Diseño de arquitectura

Para cumplir con las especificaciones solicitadas por el cliente se planificó una arquitectura de sistema distribuido.

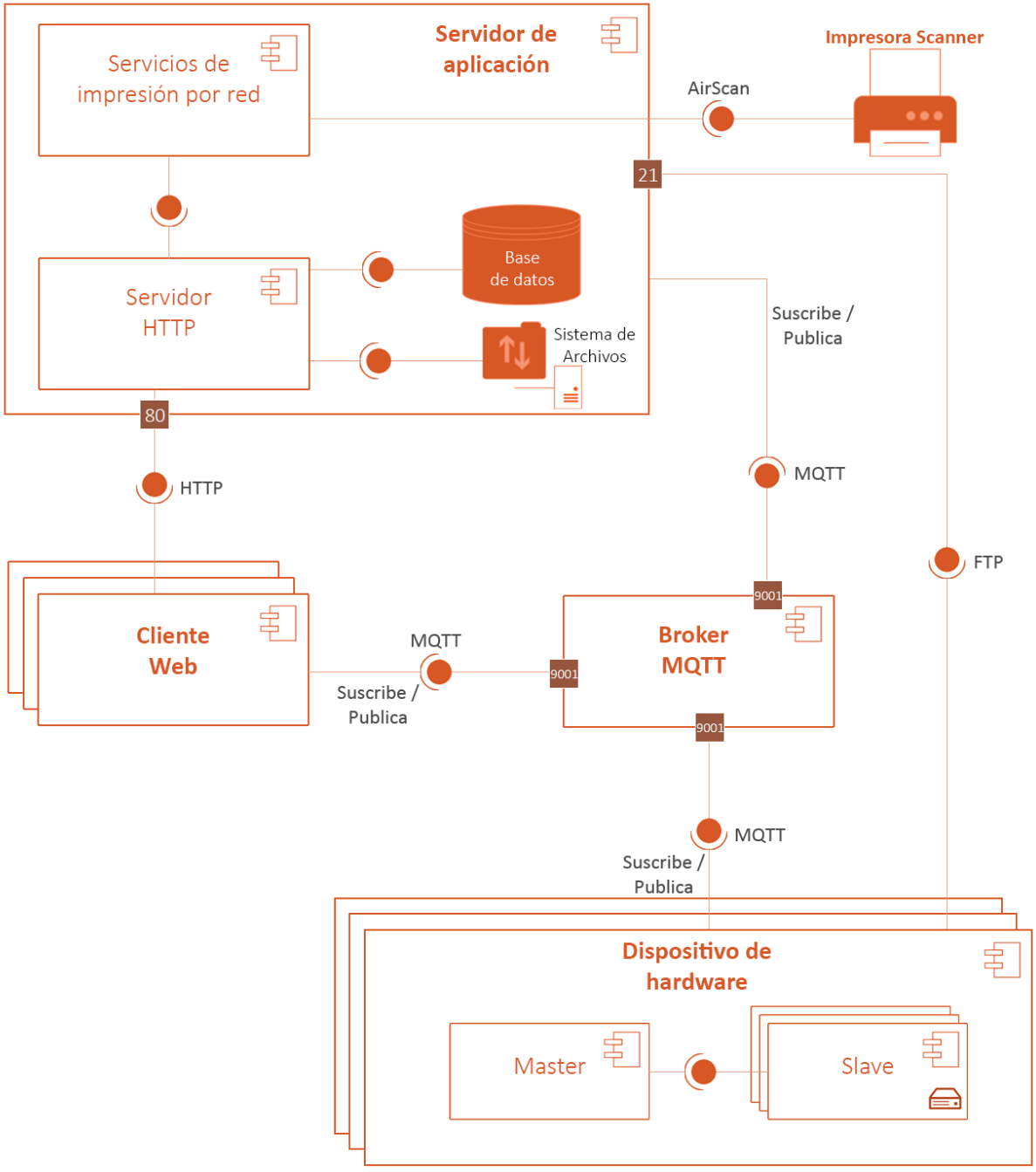


Figura 7.1 - Diagrama de componentes de arquitectura

Como se muestra en la *Figura 7.1*, la arquitectura está compuesta por:

- un servidor de aplicación, responsable de brindar los distintos servicios al resto de componentes, así como de la conexión con los dispositivos de escaneo,
- un conjunto de dispositivos de hardware encargados de la lectura de tarjetas por aproximación y el control de las barreras,
- un conjunto de clientes web que permiten a los usuarios del sistema interactuar con la información,
- y un broker MQTT que sirve como capa de abstracción para comunicar eventos entre las componentes de distinta naturaleza.

Entre los distintos servicios brindados por el servidor de aplicación se encuentra un servidor HTTP, el cual es utilizado por los distintos clientes web para acceder a la aplicación de gestión. Otro de estos servicios es un servidor FTP, el cual permite a los dispositivos de hardware actualizar sus bases de datos interna diariamente durante las horas de menor carga del servidor. De este modo, los dispositivos pueden cumplir su función incluso ante una indisponibilidad eventual de la red local.

Tanto el servidor de aplicación, como los dispositivos de hardware y los clientes web se suscriben al broker MQTT para escuchar los eventos del sistema; pero únicamente los dos primeros son quienes están habilitados para publicar nuevos mensajes.

Clientes web

Como se ha nombrado en secciones anteriores, algunos de los principales requerimientos no funcionales del sistema fueron su fácil instalación y su compatibilidad con distintos sistemas operativos. Gracias a la arquitectura del sistema, no es necesaria la instalación de drivers especializados para ninguna de sus componentes, por lo que la interacción de los clientes con el servidor de aplicación se realizará netamente a través de protocolos de red.

Debido a estas consideraciones, se optó por desarrollar los distintos clientes del sistema como aplicaciones web que puedan ser ejecutadas en los navegadores web

modernos. Como consecuencia, la instalación de un nuevo cliente resulta trivial, sólo es necesario instalar un navegador web compatible y acceder a la aplicación.

Por otra parte, los navegadores web cuentan con distintas versiones para la mayoría de sistemas operativos de la actualidad, por lo que el cliente web se abstrae del sistema operativo en el cual es ejecutado; solo requiere un navegador compatible.

Dispositivos de hardware

Cada puesto de trabajo presenta una determinada cantidad de sensores, llamados slaves. Todos estos sensores son controlados por un único dispositivo, llamado master. Este arreglo posibilita a los slaves a tener menor poder de procesamiento, siendo el master el encargado en realizar aquellas tareas más demandantes.

La lectura realizada por el sensor es comunicada al máster, siendo la responsabilidad principal del mismo. Una vez que el máster posee la información de esta lectura, se encarga de comunicar al resto de los interesados. Como contrapartida, cuando otro dispositivo, fuera del arreglo, quiere interactuar con alguno de los slaves, debe comunicarse con el máster, y es este último quien finalmente le comunica al slave.

Para cumplir con el requerimiento de que los dispositivos puedan trabajar por un tiempo prudencial aún cuando la red local no se encuentre disponible es necesario dotarlos de una base de datos interna. Esta base de datos deberá contener la información de aquellas tarjetas que se encuentran habilitadas para acceder por el puesto donde se encuentra el dispositivo. Como esta información debe ser almacenada en la memoria interna, que es una tarjeta SD de capacidad limitada, se decidió utilizar una estructura de archivo en formato CSV (del inglés *comma-separated values*).

Cada línea del archivo representa la información de una tarjeta habilitada utilizando el siguiente formato:

```
FC;ID;Hi1;Hf1;Hi2;Hf2;L;Ma;Mi;J;V;S;D;Puesto;Nivel
```

- **FC**: número entero que representa el facility code de la tarjeta. Puede tomar valores desde 0 a 255.

- **ID**: número entero identificatorio de la tarjeta. Puede tomar valores desde 0 hasta 65535.
- **Hi1:** hora inicio del rango horario habilitado para el ingreso. Los valores son en formato HHMM (siendo HH la hora en formato de 24 horas y MM los minutos).
- **Hf1:** hora finalización del rango horario habilitado para el ingreso. Los valores son en formato HHMM. Si el rango horario finaliza al día siguiente, este campo tomará el valor de "2359".
- **Hi2:** si el primer rango horario sobrepasa las 23:59, este campo se completa con el inicio del rango del día siguiente "0000" Los valores son en formato HHMM. Si no presenta segunda franja horaria toma valor "xxxx".
- **Hf2:** hora finalización de la segunda franja horaria. Los valores son en formato HHMM. Si no presenta segunda franja horaria toma valor "xxxx".
- **L:** es uno o cero dependiendo si está habilitado el ingreso el día lunes.
- **Ma:** es uno o cero dependiendo si está habilitado el ingreso el día martes.
- **Mi:** es uno o cero dependiendo si está habilitado el ingreso el día miércoles.
- **J:** es uno o cero dependiendo si está habilitado el ingreso el día jueves.
- **V:** es uno o cero dependiendo si está habilitado el ingreso el día viernes.
- **S:** es uno o cero dependiendo si está habilitado el ingreso el día sábado.
- **D:** es uno o cero dependiendo si está habilitado el ingreso el día domingo.
- **Puesto:** número de puesto por el que se encuentra habilitado. FF indica todos los puestos del área.
- **Nivel:** nivel de ingreso.

El archivo, generado por el servidor, está ordenado por número de tarjeta (FC más ID). De este modo, la búsqueda la realiza en forma secuencial.

Para reducir los tiempos de búsqueda en los dispositivos, se decidió por organizar los archivos en una estructura de árbol tipo B-Tree. Primero se accede a un archivo índice, que contiene los rangos de números de tarjetas y la referencia al archivo donde se encuentra la información de ellas. Luego, se accede secuencialmente al segundo archivo hasta encontrar la primera ocurrencia de la tarjeta. Para tal fin, cada archivo CSV puede tener, como máximo, 500 líneas. Cuando esta cantidad es superada, deberá generarse un nuevo archivo en el cual registrar el excedente de líneas; esto

puede repetirse la cantidad de veces que sean necesarias. El formato del archivo de índice es el siguiente:

```
Archivo;FCi;IDi;FCf;IDf
```

- **Archivo:** nombre del archivo que contiene la información de las tarjetas contenidas en el rango especificado.
- **FCi:** número entero que representa el facility code de la primer tarjeta contenida por el archivo. Puede tomar valores desde 0 a 255.
- **IDi:** número entero identificatorio de la primer tarjeta contenida por el archivo. Puede tomar valores desde 0 hasta 65535.
- **FCf:** número entero que representa el facility code de la última tarjeta contenida por el archivo. Puede tomar valores desde 0 a 255.
- **IDf:** número entero identificatorio de la última tarjeta contenida por el archivo. Puede tomar valores desde 0 hasta 65535.

Se asume que el índice no superará las 500 líneas ya que el sistema no contará con un número de tarjetas lo suficientemente alto como para causar esta situación. Cada vez que un dispositivo actualice su base de datos interna, deberá descargar desde el servidor de aplicación el índice junto a todos los archivos de tarjetas generados; esta descarga se realiza utilizando el protocolo FTP.

Siendo un ambiente laboral en el cual se están modificando los datos de acceso constantemente, puede requerirse habilitar o inhabilitar una tarjeta, sin necesidad de actualizar el total del conjunto de datos que posea el dispositivo. Para estas situaciones, fue diseñado un archivo de novedades, en el cual cada línea representa una tarjeta, con el mismo formato antes explicado. En este tipo de archivos pueden encontrarse entradas con tarjetas inhabilitadas, debido a que un trabajador pueda ser dado de baja y deba ser acreditado en el momento. El dispositivo es alertado de la modificación a través de un mensaje MQTT (mensaje de *actualización de tarjeta*).

Por último, cada dispositivo debe llevar un registro de todas las lecturas realizadas en un archivo determinado para tal fin, llamado *archivo de bitácora*. Se

genera un archivo de bitácora por día, siendo la fecha el nombre utilizado para este archivo; cada línea representa una lectura utilizando el siguiente formato:

```
FC;ID;Ac;Hora
```

- **FC**: número entero que representa el facility code de la tarjeta. Puede tomar valores desde 0 a 255.
- **ID**: número entero identificatorio de la tarjeta. Puede tomar valores desde 0 hasta 65535.
- **Ac**: indica si la apertura fue satisfactoria o no, tomando el valor "OK" o "NOT" respectivamente.
- **Hora**: hora en el que la lectura fue efectuada. Los valores son en formato HHMM (siendo HH la hora en formato de 24 horas y MM los minutos). El día, mes y año se encuentran registrados en el nombre del archivo.

Al momento de la lectura de una tarjeta, el primer paso es buscar en el archivo de novedades, el cual contiene la última actualización que ese número de tarjeta presenta. En caso de no encontrar a la tarjeta se procede a buscar, en el archivo índice de tarjetas, el rango de tarjetas en el cual está comprendida y, así, el archivo en el que debe estar, en caso de estar habilitada. Por último, se busca secuencialmente en el archivo antes mencionado hasta localizar a la tarjeta. Si la encuentra, utiliza la información referida a los días y horarios en los que puede acceder el trabajador para determinar si se encuentra habilitado o no. En caso de no encontrarla, se determina que el trabajador no se encuentra habilitado. Finalmente, el dispositivo genera una nueva línea en el archivo de bitácora con el resultado de la lectura.

Protocolo MQTT

Como ya se ha dicho, para la comunicación entre las distintas componentes del sistema se utilizó la tecnología MQTT (del inglés *Message Queuing Telemetry Transport*). MQTT es un protocolo de mensajería de publicación/suscripción diseñado para ser utilizado en redes inestables con poco ancho de banda y latencia alta. Estos principios lo convierten en un protocolo ideal para ser utilizado en el emergente mundo de IoT (del

inglés *Internet of Things*), donde el ancho de banda y la batería de los dispositivos son dos de los mayores limitantes.

El componente principal de este protocolo es una capa intermedia, llamada broker, que se encarga de recibir los mensajes de los distintos clientes y entregarlos aquellos clientes que tienen interés en ellos.

Este protocolo está basado en el principio de publicación de mensajes y suscripción a temas. Un conjunto de clientes se conectan a un broker, suscribiéndose a los temas que son relevantes para ellos. A su vez, un conjunto de clientes se conectan al broker para publicar mensajes en uno o más temas específicos. De esta manera, los suscriptores obtienen la información que consideran relevante sin la necesidad de conocer quién es el generador de dicha información.

Los distintos temas son tratados como una jerarquía de árbol, utilizando la barra (/) como separador. Por ejemplo, el tema "a/b/c". Los clientes pueden recibir mensajes suscribiéndose a los distintos temas. Una suscripción puede ser a un tema específico, en el cual es necesario especificar la jerarquía completa (ej. "a/b/c"), o puede ser hasta un cierto nivel de la jerarquía, utilizando comodines. Hay dos comodines "+" o "#". El primero es para indicar la suscripción a todos los temas del nivel anterior en la jerarquía. Por ejemplo, con el tema "a/+/c" el cliente se suscribiría a todos los temas que empiecen con "a" en el primer nivel y tengan a "c" en el tercer nivel. Con el segundo comodín ("#"), el cliente se puede suscribir a todos los niveles restantes en la jerarquía. Por ejemplo, con el tema "a/b/#" se reciben todos los mensajes que en el primer nivel tengan "a" y en el segundo nivel "b", sin tener en cuenta la cantidad de niveles que posea el tema ("a/b/c" y "a/b/c/d" son temas válidos).

Otro aspecto que vale la pena recalcar es el QoS (del inglés *Quality of Service*), el cual determina que tan seguro es que un mensaje haya sido recibido por un cliente determinado. Se definen tres niveles de QoS, siendo:

- Nivel 0: el mensaje es enviado una única vez y no se espera a recibir una confirmación de recepción.

- Nivel 1: el mensaje es enviado las veces que sean necesarias hasta recibir una confirmación de recepción.
- Nivel 2: el mensaje es enviado una única vez utilizando un "handshake" de cuatro etapas, asegurándose de esta manera su correcta recepción.

Niveles más altos ofrecen una confiabilidad superior pero requieren ancho de banda mayores y niveles de latencia bajos, lo que va encontra de las características de diseño principales de MQTT.

Definición de estructuras

Los dispositivos de hardware son desarrollados por el grupo de electrónica de la Facultad de Ingeniería de la UNMDP de manera independiente. Debido a esto, es necesario definir los tipos y estructuras de los mensajes que serán enviados, así como la jerarquía de los temas en los cuales estos mensajes serán publicados, para poder realizar ambos desarrollos en paralelo.

Los mensajes serán publicados en una rama principal "cp" y en el siguiente nivel será especificado el identificador del dispositivo. En un último nivel, se presentan dos canales "s" y "r", que serán utilizados para indicar una dirección a los mensajes. De este modo, los temas tienen la siguiente forma:

```
/cp/<id_dispositivo>/s/
```

```
/cp/<id_dispositivo>/r/
```

El primer canal (terminado en "/s") es utilizado para que los sensores de tarjetas envíen la información, mientras que el segundo (terminado en "/r") es en el que se suscriben para recibir aquellos datos de su interés. Por el otro lado, tanto los clientes web como el servidor se suscriben al canal terminado en "/s" y publican en el canal terminado en "/r". De esta manera se garantiza la direccionalidad de los mensajes.

Un canal más fue definido con el fin de emitir mensajes del tipo de alarma.

```
/cp/alarmas/
```

Dado que los dispositivos de hardware no cuentan con un gran poder de procesamiento, existe un limitante de 256 caracteres en cada mensaje que se envía. Debido a esto, es necesario definir una estructura de mensajería ligera para que se pueda dar respuesta al gran número de mensajes que circularán por el sistema. Para tal fin, se decidió que cada mensaje sea un JSON donde las claves están compuestas por una abreviación de la palabra (generalmente un único carácter en mayúscula), para así reducir el tamaño total del mensaje. Como mínimo, todo mensaje debe incluir el nombre del comando que representa y la fecha de emisión del mismo:

```
{
    "C": "XXXX",
    "F": "XXXX"
}
```

- **C:** nombre del comando. Cadena de caracteres que indica la orden representada por el mensaje.
- **F**: timestamp del momento de generación del mensaje.

Mensajes del servidor de aplicación

El *ping* es un mensaje que el servidor envía a un dispositivo para conocer su estado actual. Como respuesta el dispositivo debe enviar el mensaje *pong*.

```
{
    "C": "PING",
    "F": "XXXX"
}
```

El mensaje de *cambio de control de acceso* indica a todos los dispositivos que el consorcio opera bajo un nuevo nivel de riesgo.

```
{
    "C":"NIVEL",
    "F": "XXXX",
    "Nv": 1 | 2 | 3
}
```

- **Nv:** número entero que determina el nivel de riesgo del consorcio. Puede tomar valores del 1 al 3.

El mensaje de *recarga de base de datos interna* indica a un dispositivo que debe actualizar los datos de las tarjetas que tiene almacenado en su memoria interna. Como respuesta, el dispositivo deberá descargar la nueva base de datos utilizando el protocolo FTP.

```
{
    "C": "RECARGA",
    "F": "XXXX"
}
```

El mensaje de *actualización de tarjeta* indica a un dispositivo que debe actualizar la información sobre una tarjeta particular en su base de datos interna.

```
{
    "C": "NOVEDAD",
    "F": "XXXX",
    "NL": "XXXX"
}
```

- **NL:** información actualizada de la tarjeta, respetando el formato CSV especificado anteriormente.

El mensaje de *envío de bitácora* indica a un dispositivo que debe cargar en el servidor sus archivos con el histórico de lectura de tarjetas. Como respuesta, el dispositivo debe subir dichos archivos mediante FTP y reiniciarlos localmente, para liberar el almacenamiento interno.

```
{
    "C": "BITACORA",
    "F": "XXXX"
}
```

El mensaje de *solicitud de sincronismo de tiempo* es envíado a un dispositivo con el fin de que este último adopte el tiempo del servidor para sincronizarse.

```
{
    "C":"SYNC",
    "F": "XXXX"
}
```

Mensajes de dispositivos

Además de los datos mínimos ya indicados, un mensaje generado por un dispositivo debe incluir la información identificatoria del emisor:

```
{
      "N": "XXXX",
      "P": "XX",
}
```

- **N**: nombre del master emisor. Cadena de 6 caracteres única para cada máster.
- **P**: identificación del slave emisor. Cadena de 2 caracteres única para cada slave en un máster determinado.

El mensaje de *inicio de dispositivo* indica el encendido de un dispositivo. Este mensaje es enviado cada vez que un dispositivo es reiniciado.

```
{
      "C": "INICIO",
      "F": "XXXX",
      "N": "XXXX",
      "P": "XX"
}
```

El mensaje de *apertura de puerta* indica la lectura de una tarjeta de aproximación por uno de los sensores. La apertura puede ser satisfactoria o no, dependiendo del estado actual de la tarjeta dentro del sistema.

```
{
      "C": "APERTURA",
      "F": "XXXX",
      "N": "XXXX",
      "P": "XX",
      "FC": XXX,
      "ID": XXXXX,
      "Ac": "OK" | "NOT"
}
```

- **FC**: número entero que representa el facility code de la tarjeta. Puede tomar valores desde 0 a 255.
- **ID**: número entero identificatorio de la tarjeta. Puede tomar valores desde 0 hasta 65535.
- **Ac**: indica si la apertura fue satisfactoria o no, tomando el valor "OK" o "NOT" respectivamente.

El mensaje de *pong* sirve como respuesta a una solicitud de *ping* solicita por el servidor de aplicación. Contiene el estado actual del dispositivo.

```
{
      "C": "PONG",
      "F": "XXXX",
      "N": "XXXX",
      "P": "XX",
      "FA": "XXXX",
      "UFID": "XXXX",
      "UFC": XXX,
      "UID": XXXXX,
      "UAc": "OK" | "NOT"
}
```

- **FA**: timestamp del momento de arranque del dispositivo.
- **UFID**: timestamp del momento de la última lectura.
- **UFC**: número entero que representa el facility code de la última tarjeta leída. Puede tomar valores desde 0 a 255.
- **UID**: número entero identificatorio de la última tarjeta leída. Puede tomar valores desde 0 hasta 65535.

- **UAc**: indica si la última lectura fue satisfactoria o no, tomando el valor "OK" o "NOT" respectivamente.

El mensaje de *alarma silenciosa* es enviado por el dispositivo cuando el usuario acciona el botón manual con el fin de alertar a los demás puestos de trabajo sobre alguna situación de interés.

```
{
    "C": "ALARMA",
    "F": "XXXX",
    "N": "XXXX",
    "P": "XX"
}
```

Mensajes de clientes web

El mensaje de *alarma silenciosa* es enviado por el cliente web cuando el usuario presiona el botón que tiene la interfaz con el fin de alertar a los demás puestos de trabajo sobre alguna situación de interés.

```
{
    "C": "ALARMA",
    "F": "XXXX",
    "U": X,
    "Z": X
}
```

- **U**: identificador del usuario del sistema que emite la alarma.
- **Z**: identificación de la zona de trabajo desde donde se emite la alarma.

8. Implementación

Tecnologías utilizadas

Al momento de implementar el sistema se tenía la libertad de configurar el servidor para que la aplicación pueda funcionar de la manera más óptima posible. Para el sistema operativo se eligió una versión orientada a servidores de *Debian*, una distribución GNU/Linux basada en software libre. El mismo otorga una gran estabilidad, seguridad y soporte de hardware. No obstante, el sistema podrá ser instalado sobre cualquier otro sistema operativo que cumpla con los requisitos detallados más adelante.

Sobre el sistema operativo se brindan los distintos servicios necesarios para el correcto funcionamiento del sistema. Cuando más de una tecnología presentaba un nivel similar de calidad, se optó por aquella que es una solución de software libre, debido a la comunidad que la sustenta, el gran soporte existente y las libertades de licencia que ofrecen.

Como servidor HTTP se utiliza *Apache*, el servidor HTTP más elegido a nivel mundial. La elección se debe a que posee una gran documentación, compatibilidad con otras tecnologías, robustez, seguridad y escalabilidad. Este servidor web es de fácil instalación y configuración, otorgando una gran flexibilidad para adaptarlo a nuestras necesidades. Una gran ventaja de utilizar esta tecnología es que permite a los distintos clientes acceder a través de un navegador, sin necesidad de una previa instalación del sistema.

El servidor de aplicación fue desarrollado, en su mayoría, utilizando el lenguaje de programación PHP. Además de que ya se contaba con experiencia utilizando el lenguaje, se optó por este lenguaje debido a la gran cantidad de documentación existente, la simplicidad de comunicación con la base de datos, el fácil acceso a paquetes que se integran con PHP y la baja curva de aprendizaje que presenta. Como base se optó por el framework de desarrollo *Laravel* en su versión 5.6, ya que ofrece soluciones robustas a las problemáticas comunes en todo sistema web, además de buenas prácticas

en seguridad y un gran uso de patrones de programación que da como resultado un código organizado y entendible. Los requisitos mínimos para su instalación son:

- PHP en su versión 7.1.3 o posterior
- Composer
- Extensión OpenSSL de PHP
- Extensión PDO de PHP
- Extensión Mbstring de PHP
- Extensión Tokenizer de PHP
- Extensión XML de PHP
- Extensión Ctype de PHP
- Extensión JSON de PHP

Otra característica por la cual se optó por la utilización de este framework es su capa de abstracción de base de datos llamada *Eloquent ORM*. La misma ofrece una representación de cada tabla a través de una clase de PHP, llamada Modelo, simplificando el acceso de datos mediante funciones en lugar de lenguaje SQL.

Como servidor de base de datos se consideró la utilización de los motores *MariaDB* y *MySql*, inclinándose por este último debido a que, al momento de inicio del desarrollo, el motor *MariaDB* no contaba con soporte para columnas de tipo JSON, las cuales resultan necesarias para simplificar ciertas queries.

A la hora de desarrollar las interfaces de usuario de los clientes web se decidió la utilización de la librería *Vue.js*, la cual está basada en JavaScript, HTML y CSS. Esta librería permite la creación de complejas páginas web SPA (del inglés *Single Page Application*) de manera progresiva y consistente. Las páginas SPA son la tendencia actual para ofrecer soluciones web equiparables a aplicaciones de escritorio, brindando mayor fluidez y menor tiempo de respuesta al momento de trabajar. Como base para las distintas componentes visuales se utilizó *Bootstrap*, librería más extendida y con mayor soporte para tal fin. Ambas tecnologías utilizan la nueva versión *ECMAScript* 6, la cual no es soportada por la mayoría de navegadores actuales. Para solventar este impedimento, se utiliza una herramienta desarrollada en *Node.js*, llamada *webpack*, para

compilar el código a una versión anterior de ECMAScript soportada por todos los navegadores modernos.

Por otra parte, para comunicar a los clientes web con el servidor de aplicación de manera asincrónica se utilizó la librería *axios* de JavaScript. La misma permite realizar llamadas AJAX de manera simple y eficaz.

El sistema tiene como requisito poder utilizar las impresoras dentro del mismo. Para imprimir, se utilizó *CUPS* (del inglés *Common UNIX Printing System*), siendo un potente software para imprimir desde las diferentes aplicaciones como desde el propio navegador. Otro requisito es la posibilidad de escanear desde el navegador web, función no soportada por ningún navegador. Para cumplir esta función fue necesario utilizar *AirScan*, una solución emergente que permite comunicarse con un escáner a través del protocolo HTTP para así solicitar un nuevo escaneo y recibir la imágen resultante del mismo.

La tecnología *HLS.js* fue seleccionada para visualizar videos en vivo a través de HTTP. Este requiere un navegador con soporte para *HTML5 video* y la API *Media Source Extenions* para funcionar.

Finalmente, para el broker MQTT se optó por *Eclipse Mosquitto*, ya que es una solución ligera que ofrece todo lo necesario para implementar los requerimientos ya descritos. Para los clientes web se utilizó la solución *Paho MQTT*, también de Eclipse, ya que es la más completa y con mejor documentación al momento.

Módulo de administración

El ciclo de vida laboral de un trabajador, así como de la empresa, inicia con su inscripción en la oficina de administración del CPRMDP, por lo que el sistema cuenta con un módulo específico para este sector. La tarea principal de los usuarios de este módulo es la gestión de la información relacionada a todos aquellos trabajadores que tienen permitido circular dentro de las áreas del puerto que son consideradas como zonas seguras. Para tal fin, el sistema debe proveer una forma sencilla de mantener actualizada la información personal de cada individuo, así como relacionarla con la

información correspondiente a las demás entidades que juegan un papel clave a la hora de permitir o denegar un ingreso.

Con motivo de diseño de datos se realizó un Modelo de Entidad-Relación (MER) para comprender cómo interactúan las distintas entidades dentro del sistema en este módulo, como se observa en la *Figura 8.1*.

La entidad que representa a un trabajador dentro del sistema es *Persona*, la cual cuenta con un número de CUIL o DNI para indicar su unicidad y demás información personal. La misma puede estar relacionada con una *Empresa* o muchas, denominando a esta relación como *Trabajo*. Existen casos en los que el trabajador no depende de una empresa, sino que trabaja de manera autónoma. De esta forma, la persona no tiene una empresa asociada pero sí existe el trabajo, a fines de que todos ellos puedan ser tratados de manera homogénea.

Una vez definido, al trabajo se le deben asociar otras dos entidades que sirven para definir si el mismo se encuentra habilitado o no para el ingreso. La primera de estas entidades es la *Tarjeta*, representación dentro del sistema de las tarjetas por aproximación que cada trabajador posee para ingresar al puerto; como mínimo, un trabajo debe tener una tarjeta asociada, pudiendo esta cantidad crecer en base a las necesidades particulares de cada trabajador. Cada tarjeta posee un número identificatorio único y un rango de fechas que determinan su validez; en base a esta información, se puede determinar si la misma se encuentra vigente al momento del ingreso.

La otra entidad que debe ser relacionada a un trabajo es el *Grupo*, el cual se asocia a una o muchas *Zonas* (que representa a cada una de las zonas restringidas del puerto), con los días y horarios en los cuales dicho trabajo puede ser ejercido. De manera opcional, un grupo puede tener una empresa asociada. Si esto ocurre, dicho grupo solamente puede ser asociado a los trabajos de los trabajadores de la empresa indicada; en caso contrario, el grupo es válido para cualquier trabajo del sistema. En definitiva, el grupo tiene la información del lugar y horarios habilitados.

Adicionalmente, cada trabajo debe tener asociado una *Actividad*, dentro de las estipuladas por el CPRMDP, la cual describe la condición bajo la cual el trabajador ingresa al puerto.

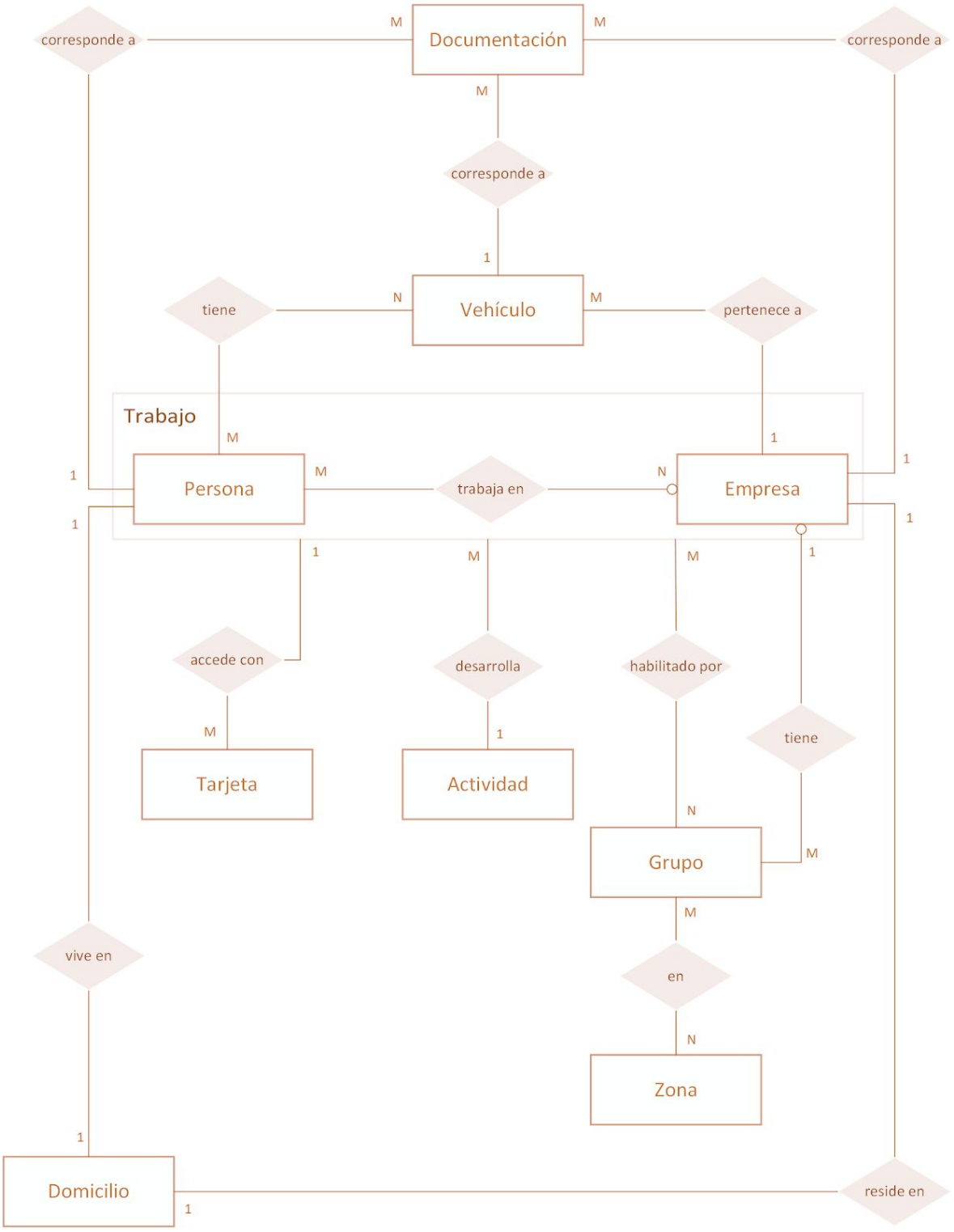


Figura 8.1 - Modelo entidad-relación del módulo de administración

Tanto los trabajadores como las empresas son asociadas a un *Domicilio*. A su vez, una empresa posee un número de CUIT único que la identifica y más información.

Las empresas tienen asociados los *Vehículos* que están habilitados a ingresar a las zonas restringidas. Así mismo, para que un trabajador pueda ingresar utilizando un vehículo, debe existir una asociación que lo habilite.

Para habilitar a trabajadores, empresas o vehículos, existe cierta *Documentación* que el responsable debe presentar al momento de inscribirse. La existencia o no de estos documentos son los que habilitan o no a las distintas entidades a poder ingresar al área de interés. Usualmente, los documentos tienen vencimiento, lo que indica que el responsable debe renovarlos para seguir conservando el estado de habilitado.

Sobre estas relaciones se desarrollan las distintas funciones que los usuarios del módulo pueden realizar.

La tarea de los usuarios es mantener actualizada la información intrínseca de los trabajadores en las bases de datos del sistema. Esto implica inscripciones de trabajadores nuevos, actualización de datos y de documentación de trabajadores ya existentes, asociaciones con vehículos y empresas previamente dados de alta y renovaciones de tarjetas. La información actualizada de las empresas y vehículos habilitados también es mantenida por estos usuarios.

Se presentan listados con buscadores para poder filtrar la información y realizar búsquedas exhaustivas de interés con el fin generar reportes. Todos estos listados pueden ser exportados a un formato de hoja de cálculos compatible con Microsoft Excel.

Cada una de estas entidades tiene documentación que expira luego de un cierto tiempo (por ejemplo seguros, habilitaciones, etc.). Debido a esto, el módulo cuenta con la funcionalidad para conocer la documentación ya vencida y a la espera de una renovación, o próxima a vencer para informarle al trabajador o empresa. El vencimiento de la documentación implica un impedimento al momento de ingreso, si así lo expresa el usuario de administración.

Si se considera que en un futuro la documentación particular de una entidad será necesaria, existe un apartado que permite a los usuarios escanear dicho documento y

almacenarlo digitalmente dentro del sistema para ser consultado en cualquier momento.

Tanto trabajadores como empresas o vehículos pueden ser inhabilitados para su ingreso en cualquier momento por decisión de un usuario. Una inhabilitación de una empresa implica la inhabilitación de todos los trabajadores que pertenezcan a la misma. Si un trabajador se encuentra asociado a más de una empresa, podrá seguir ingresando con aquellos trabajos que cumplan las condiciones necesarias.

Otra función importante es la corroboración del estado del pin (número de la tarjeta). Por la naturaleza de su trabajo, los guardias tienen una cantidad reducida de información sobre el estado de un pin. Cuando se requiere más información sobre por qué pudo haber sido denegado un pin, los usuarios de administración son que pueden acceder a esta información mediante una sección específica del módulo. Un pin está habilitado si cumple con los siguientes requisitos:

- Trabajador habilitado
- Empresa habilitada
- Trabajo (relación trabajador con empresa) habilitado
- Documentación vigente
- Fecha de ingreso previa al vencimiento de la tarjeta
- Tarjeta activa (pueden darse de baja por robo, pérdida o rotura)
- Nivel de riesgo habilitado del trabajador correspondiente con el nivel de riesgo actual.

Las renovaciones de tarjetas de los trabajadores se realizan actualizando el vencimiento en sistema e imprimiendo una nueva plantilla generada por el sistema con el nuevo vencimiento. También puede darse el caso de que se extravíe una tarjeta, lo que requiere darla de baja para que ningún otro trabajador pueda entrar con la misma.

Por último, existe un sistema de observaciones sobre las distintas entidades mencionadas que permite una comunicación interna entre los usuarios de los distintos módulos del sistema. Estas observaciones permiten llevar registro de anomalías o eventos que no son registrados en el flujo normal de información.

Módulo de guardias

Una vez el trabajador se encuentra inscripto en el sistema, su interacción principal con el mismo será a través de los distintos accesos a las zonas restringidas. Para lograr esto se plantean las siguientes relaciones entre entidades:

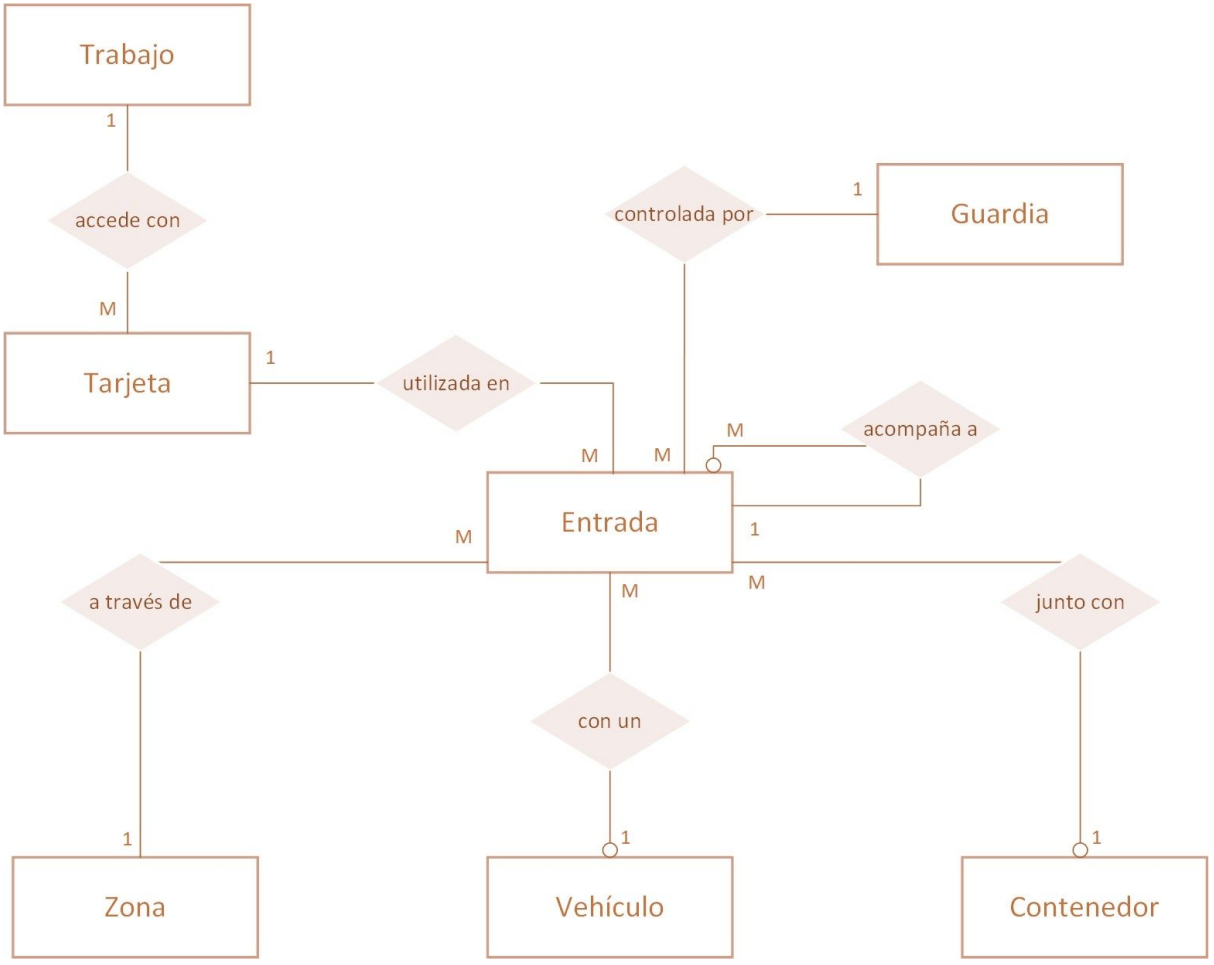


Figura 8.2 - Modelo de entidad-relación del módulo de guardias (entradas convencionales)

Consideraciones adicionales:

- Si la *Entrada* posee un *Vehículo*, el mismo debe estar asociado a la *Persona* propietaria de la *Tarjeta* con la que se accede.
- Si la *Entrada* posee un *Contenedor*, la mismas debe poseer un *Vehículo* habilitado a transportar contenedores.

- Si una *Entrada* acompaña a otra entrada, esta última no puede ser acompañante de ninguna otra.

La entidad principal es la *Entrada*, la cual almacena toda la información relacionada a un ingreso (o egreso) de un trabajador a través de una zona restringida del puerto. Toda entrada se realiza utilizando una *Tarjeta*, la cual debe estar habilitada al momento de su uso. Esta tarjeta permitirá conocer a que *Trabajo* (relación *Persona* - *Empresa* de la *Figura 8.1*) corresponde el ingreso y, en consecuencia, conocer el trabajador que lo realizó.

Además de la tarjeta, toda entrada debe ser realizada a través de una *Zona* existente. Para que el ingreso sea satisfactorio, el trabajo debe estar asociado a la zona a la que se intenta ingresar mediante una relación con un *Grupo* (como se muestra en la *Figura 8.1*). A su vez, la entrada debe ser controlada por un *Guardia*, que es la representación en el sistema de los usuarios que utilizan este módulo.

Si la entrada se realiza a través de un acceso vehícular, la misma deberá ser asociada con el *Vehículo* que se utilizó al ingresar, el cual debe estar asociado a la *Persona* que está realizando el ingreso. En ocasiones en las que también se ingrese junto a un *Contenedor*, el vehículo utilizado debe estar habilitado a transportar contenedores.

Existen casos de ingresos utilizando un vehículo en los cuales el conductor viene acompañado por uno o más trabajadores. En esta situación, existirá una entrada principal, la cual posee los datos propios del conductor del vehículo. Adicionalmente, se generará una entrada por cada uno de los acompañantes, y estas entradas referenciarán a la entrada principal del conductor. En este caso, los acompañantes pueden no poseer una asociación con el vehículo mediante el cual están ingresando.

Sobre estas relaciones se desarrollan las distintas funciones que los usuarios del módulo pueden realizar.

La función principal de este módulo es el control y registro de los distintos movimientos a través de las entradas de las zonas restringidas del puerto. El proceso comienza cuando un trabajador aproxima su tarjeta personal a un lector de tarjetas ubicado en una zona restringida. Para esto se utilizan los dispositivos de hardware,

mensajes y protocolo MQTT descritos en secciones anteriores. Dicho lector obtendrá la información correspondiente a la tarjeta en su base de datos interna. Si el sistema está trabajando en modo *offline*, se utiliza dicha información para la decisión de acceso (o egreso) y el proceso finaliza en este momento, almacenando la entrada en la bitácora interna del dispositivo.

En caso de que el sistema funcione en modo *online*, el proceso continúa enviando el mensaje *apertura de puerta* a través del canal MQTT correspondiente. Una vez el mensaje es enviado, el broker MQTT se encargará de hacerlo llegar al cliente web correspondiente a la zona de ingreso. Cuando esto sucede, el cliente solicita al servidor de aplicación la información actualizada de la tarjeta, para evitar cualquier incongruencia de la base de datos interna del lector. Cuando el servidor responde, el cliente web muestra en la pantalla toda la información del trabajador relevante para la decisión.

Es aquí cuando el guardia determina si el trabajador puede o no ingresar (o egresar). Sí la decisión del guardia va en consonancia con la decisión del sistema, el proceso finaliza y la nueva entrada (o denegación de la misma) es registrada en la base de datos. Si la decisión del guardia difiere con la decisión del sistema, existe una etapa extra en la cual el guardia debe especificar la razón por la cual toma una decisión contraria; luego de esto la entrada (o denegación de la misma) es registrada normalmente.

En casos excepcionales donde el trabajador no posea su tarjeta al momento del ingreso (o egreso), el guardia tiene la capacidad de realizar una búsqueda por el número de D.N.I. de la persona. De esta manera, el servidor de aplicación brindará al guardia la información de todas las tarjetas activas del trabajador para poder seguir con el flujo de trabajo antes mencionado.

A continuación se presenta el diagrama de secuencias del sistema para esta funcionalidad:

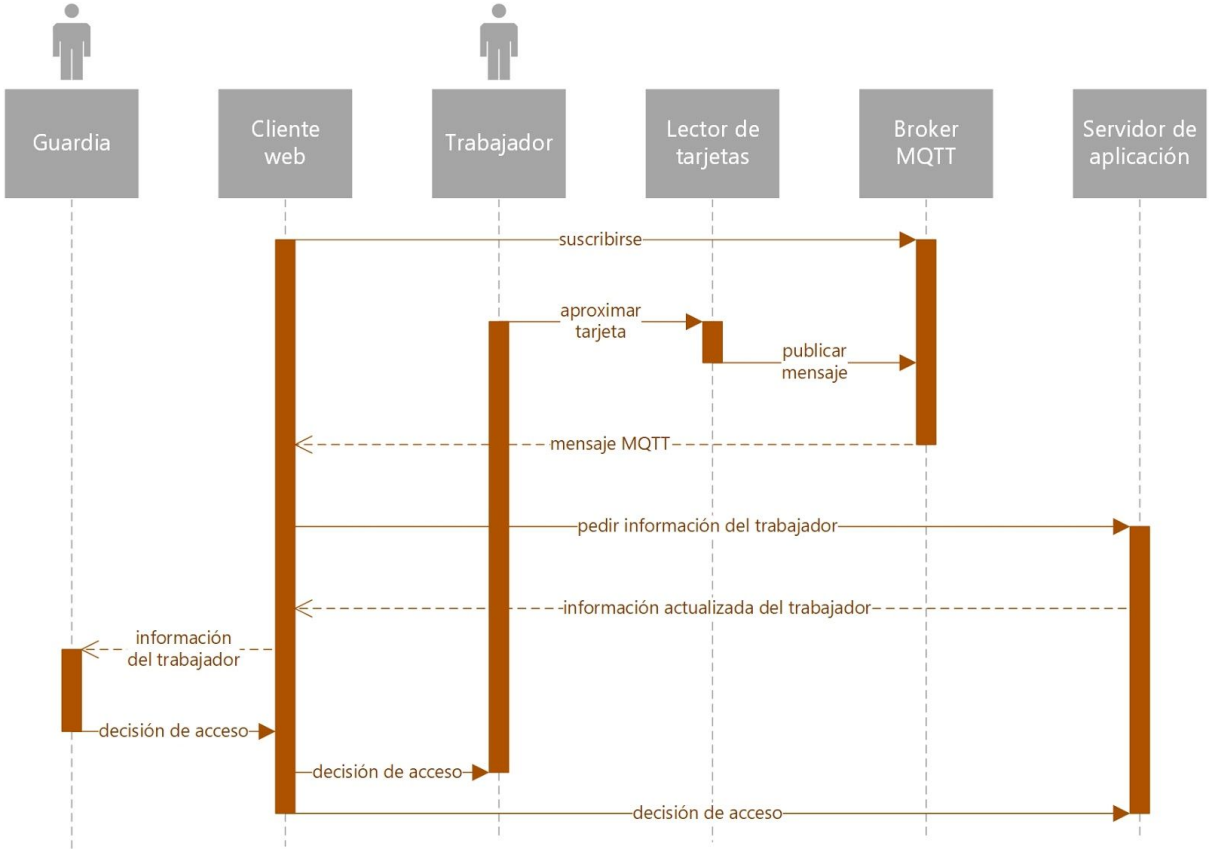


Figura 8.3 - Diagrama de secuencia del módulo de guardias

Adicionalmente a las entradas de trabajadores, existe un tipo especial de acceso para personas que no desempeñan un trabajo dentro del puerto pero que deben ingresar por algún motivo en especial. A este tipo de personas se las conoce como *invitados*. Las relaciones que definen un ingreso de este tipo son:

Figura 8.4 - Diagrama de entidad-relación del módulo de guardias (entradas de invitados)

La entidad principal es la *Entrada de invitado*, la cual almacena toda la información relacionada a un ingreso (o egreso) de un invitado a través de una zona restringida del puerto. Toda entrada se realiza utilizando el D.N.I. de la *Persona*, la cual debe tener una *Invitación* válida al momento del ingreso. Para que una invitación sea válida, la misma debe poseer una fecha de inicio menor y una fecha de finalización mayor a la fecha de ingreso.

Además de la invitación, toda entrada debe ser realizada a través de una *Zona* existente. A su vez, la entrada debe ser controlada por un *Guardia*. Este tipo de entradas sólo pueden realizarse a través de los accesos peatonales, por lo que los vehículos y contenedores no se encuentran relacionados a la misma como en el caso de un acceso de un trabajador.

El flujo de trabajo para este tipo de ingresos se encuentra simplificado en relación a las entradas de trabajadores, ya que aquí no intervienen los lectores de tarjetas. El proceso inicia cuando el invitado le proporciona su D.N.I. al guardia, quien lo ingresa manualmente en el cliente web. Este último solicita al servidor de aplicación la información actualizada sobre este invitado y, una vez obtiene la respuesta, muestra en la pantalla toda la información relevante para la decisión.

Es aquí cuando el guardia determina si el invitado puede o no ingresar (o egresar). Sí la decisión del guardia va en consonancia con la decisión del sistema, el proceso finaliza y la nueva entrada (o denegación de la misma) es registrada en la base de datos. Si la decisión del guardia difiere con la decisión del sistema, existe una etapa extra en la cual el guardia debe especificar la razón por la cual toma una decisión contraria; luego de esto la entrada (o denegación de la misma) es registrada normalmente.

Para facilitar el trabajo del guardia, el sistema cuenta con un lector de códigos de barra en aquellas terminales de trabajo donde el flujo de invitados es mayor, el cual es capaz de leer la información del D.N.I. del invitado. Una vez esta información es obtenida, el cliente se encarga de decodificarla para así obtener el número de documento de manera automática y realizar la petición al servidor sin intervención

directa del guardia.

Además de supervisar las entradas en curso, este módulo brinda a los guardias la funcionalidad de consultar el histórico de entradas, ya sea de trabajadores como de invitados, de las últimas 24 horas (límite fijado por el CPRMDP) en su puesto de trabajo. Para eso, se brinda un buscador donde pueden filtrarse las entradas por número de tarjeta, número de documento, vehículo utilizado y/o hora exacta.

Otra función de este módulo es llevar un registro de los eventos importantes que le suceden a los guardias a través del denominado *Libro del Guardia*. Cada guardia al comienzo de su turno inicia sesión en una terminal específica. En esa pantalla va a visualizar sus entradas anteriores en el *Libro del Guardia*, más las entradas que hicieron los demás guardias que hayan trabajado en esa terminal. De este modo, se genera una comunicación interna, dejando un registro de la misma.

Para tener un panorama más amplio de su puesto de trabajo, la interfaz web posee la funcionalidad de realizar streaming de video de la cámara de seguridad ubicada en la terminal del guardia.

Por último, el módulo cuenta con el mismo sistema de observaciones que el módulo de administración, sólo que limitado a personas. Debido a esto, un guardia podrá generar o consultar cualquier observación realizada sobre una persona por cualquier usuario del sistema, pero no podrá hacerlo para vehículos ni empresas.

Módulo de monitoreo

El tercer módulo está destinado a aquellos usuarios que se encargan de observar las áreas del CPRMDP. Al tener la necesidad de monitorear, se le garantiza a los usuarios los permisos de lectura sobre todas las entidades del sistema.

Las funciones principales de este módulo incluyen la visualización y filtrado sobre los listados de personas, empresas y vehículos, así como la información detallada de cada uno. Además, se podrán generar reportes para realizar informes sobre las

búsquedas de interés que se realizan. También tienen la posibilidad de ver el estado de un pin, si está habilitado o no, y las razones por las cuales puede no estarlo.

Así mismo, los operadores de monitores tienen acceso a la información de todas las terminales de los guardias. En primer lugar, pueden consultar el *Libro del Guardia* para conocer los eventos importantes reportados por los distintos usuarios. En segundo lugar, los operadores tienen acceso al listado histórico de entradas (los guardias solo visualizaban las entradas de las últimas 24 horas) que suceden en todo el puerto, pudiendo realizar búsquedas exhaustivas para generar reportes.

Por último, así como todos los usuarios, y con el fin de poder generar comunicación entre cada uno de ellos, pueden realizar observaciones sobre personas, empresas o vehículos. Este, es el único permiso de escritura sobre el sistema de este tipo de usuarios.

Módulo de gerencia

El último módulo del sistema es el utilizado por la gerencia de seguridad del CPRMDP. Este módulo posee todas las funcionalidades descritas para los módulos de administración y monitoreo (incluyendo el sistema de observaciones), además de tres funcionalidades únicas.

La primera de estas funcionalidades es la gestión de usuarios del sistema. Gracias a esto, la gerencia pueden ver, crear, modificar y/o dar de baja cualquier cuenta de usuario de cualquier módulo del sistema.

La segunda funcionalidad específica de este módulo es dar de alta o modificar invitaciones para personas que no realizan un trabajo dentro del puerto pero requieren ingresar por alguna razón en específico. Una invitación cuenta con un rango de fechas durante la cual la persona podrá ingresar por cualquier ingreso peatonal, una descripción del motivo por el cual es invitado y la aceptación por parte de la persona invitada del manual de conducta dentro de las zonas restringidas del CPRMDP.

Por último, el módulo cuenta con la funcionalidad de cambiar el nivel de riesgo bajo el cual el sistema funciona. Los niveles de riesgo existentes son el Nivel 1, Nivel 2 y Nivel 3, siendo cada uno más restrictivo que el anterior. Un nivel de riesgo dado restringe el acceso a cualquier persona que no posea una tarjeta con dicho nivel de riesgo, o una con un nivel superior.

Sincronización de datos de dispositivos

Los dispositivos de hardware necesitan tener la información lo más actualizada posible, dado que si hay cambios en el sistema estos se tienen que enterar para permitir o denegar el acceso según las nuevas modificaciones.

Cuando hay una modificación sobre el estado de habilitado o deshabilitado de una persona, el servidor publica el mensaje de *actualización de tarjeta*, el cual será recibido y procesado por aquellos dispositivos interesados. Por otro lado, cuando una empresa es bloqueada o desbloqueada, debido a la gran cantidad de tarjetas que es necesario actualizar, se emite el mensaje de *recarga de base de datos interna*, el cual desencadenará la actualización de la información de todas las tarjetas del sistema, para evitar un número excesivamente elevado de mensajes MQTT que puedan desestabilizar la red y generar pérdidas.

Además de la sincronización debido a una acción de un usuario del sistema, existe una sincronización automática que se realiza diariamente. Esta actualización es la encargada de verificar los vencimientos de los distintos datos relevantes para el ingreso y actualizar la información de las tarjetas en las bases de datos internas de los dispositivos.

Otros trabajos de sincronización que se ejecutan periódicamente de manera automática son los relacionados a la sincronización de la bitácora y tiempo de cada dispositivo.

9. Seguridad

La seguridad informática es la habilidad de identificar y eliminar vulnerabilidades. Para evitar el acceso a la información del sistema por parte de personas ajenas al mismo, y para impedir que los distintos tipo de usuario puedan acceder a información que no está destinada a ellos, se aplicaron una serie de medidas de validación y seguridad que abarcan todo el sistema.

Algunas de estas medidas fueron desarrolladas por el equipo del proyecto, mientras que algunas son provistas por alguna de las tecnologías utilizadas en el desarrollo del sistema.

Cuentas y roles de usuario

Todo usuario debe contar con una única cuenta, personal e intransferible, con la que puede acceder al sistema. Esta cuenta posee un nombre de usuario y una contraseña, y son generadas por los usuarios del módulo de gerencia. Para evitar una posible filtración de las cuentas de usuario, las contraseñas se encuentran encriptadas utilizando *bcrypt*.

Además de los datos personales de su dueño, cada cuenta posee un rol que determina a qué secciones del sistema el usuario puede acceder y que datos del mismo puede modificar. Existen cuatro roles posibles, siendo ellos *administrador*, *guardia*, *monitor* y *gerente*.

Actividad

Todos los movimientos realizados por los usuarios quedan registrados. De este modo, se otorga una capa de seguridad más allá de la seguridad del software, evidenciando a los usuarios que tienen acceso al sistema y realicen posibles acciones malintencionadas, dándole a los gerentes la posibilidad de tomar acciones contra ellos.

Middlewares

Laravel provee una capa que funciona como un puente entre las peticiones realizadas y las respuestas a las mismas. Esta funcionalidad se conoce como *Middleware*. Además, ofrece una funcionalidad conocida como *Policy*, la cual permite agrupar una serie de validaciones de seguridad que los usuarios deben cumplir sobre una entidad en particular para poder acceder a los datos de la misma.

En primer lugar, se tiene un middleware de autenticación, privando a aquellos que no posean una sesión iniciada a realizar acciones sobre el sistema.

Luego, gracias a la flexibilidad que ofrece el framework para generar nuevas Policies y el sistema de roles de cuenta de usuario, se construyeron una serie de policies que aseguran que solo aquellos usuarios autorizados puedan acceder y/o modificar los recursos brindados por el servidor de aplicación.

SQL Injection y Cross-site request forgery

Dos de los ataques maliciosos más extendidos en los sistemas web son los conocidos como *SQL Injection*, que ataca a las base de datos, y *CSRF* (del inglés, Cross-site request forgery), que ataca a la autenticación del sistema. Laravel trae incorporado distintos métodos que evitan estos tipos de ataques, los cuales fueron utilizados.

En primer lugar, toda consulta realizada a la base de datos a través del ORM Eloquent se encuentra sanitizada, es decir, es validada y se remueven de ella todos aquellos caracteres especiales que podrían utilizarse para explotar una vulnerabilidad.

En cuanto a CSRF, Laravel utiliza una técnica de token único para cada usuario generado al momento del inicio de sesión por un método impredecible. Este token es enviado en cada petición que el usuario realice, lo que valida su autenticidad en el sistema.

Red de trabajo

El sistema es para uso exclusivo dentro del consorcio portuario, no siendo accesible fuera del mismo. Por esta razón, el mismo se ejecuta dentro de una red local, a la que solo tienen acceso los trabajadores del consorcio.

Gracias a esto, se brinda una capa más de seguridad, ya que los datos que viajan a través de esta red no pueden ser interceptados por personas ajenas a la misma ni son enviados a través de Internet.

Baterías UPS

El servidor cuenta con una batería UPS (del inglés *uninterruptible power supply*), la cual le permite funcionar, en caso de un corte de luz, por un tiempo finito. Así mismo, los dispositivos de hardware tienen batería UPS que les permite trabajar con independencia. De esta manera, se garantiza la disponibilidad e integridad de los datos y servicios del sistema.

10. Puesta en marcha

Al momento de instalar el sistema, primero fue necesaria la configuración del servidor donde el mismo se encuentra alojado. Una vez adquirido el hardware por parte del consorcio portuario, se procedió a la instalación y configuración del sistema operativo Debian y el resto de tecnologías utilizadas, las cuales fueron especificadas en la sección de *Implementación*. Una ventaja de contar con un servidor dedicado, el cual fue configurado íntegramente por nosotros, es la posibilidad de instalar las herramientas y versiones específicas que cumplen con nuestros requerimientos.

El sistema fue instalado por etapas para realizar las pruebas y capacitaciones necesarias, como se explicó en la sección de *Modalidad de Trabajo*. Debido a esto, durante el periodo de transición, fue necesario conservar el sistema anterior para que los operarios del CPRMDP pudiesen continuar con su trabajo hasta finalizar el nuevo desarrollo. Para tal fin, como el anterior sistema funciona sobre una versión para servidores de Windows, fue necesaria la instalación y configuración de un entorno de virtualización sobre el cual el mismo sería ejecutado.

En la instalación, y en cada actualización, del sistema es necesario realizar ciertas tareas. La primera de ellas es compilar el código de JavaScript ES6 a una versión compatible con los navegadores a utilizar. Por otro lado, hay que actualizar la base de datos, en caso de que haya una alteración. Finalmente, es necesario verificar el correcto funcionamiento de los trabajos automáticos que el servidor debe realizar.

Una vez finalizada la instalación y pruebas del módulo de administración, fue posible migrar los datos del sistema anterior al nuevo sistema para que los usuarios puedan comenzar a realizar sus actividades. Gracias a este trabajo, se evitó la necesidad de realizar un re-empadronamiento masivo de trabajadores, lo que hubiese significado un costo alto a nivel administrativo. Este módulo fue instalado en el mes de noviembre del año 2018. La elección de instalar este módulo primero fue debido a que los otros módulos trabajan con la información que se carga en este.

En los meses siguientes (diciembre de 2018 a febrero de 2019) se realizaron las instalaciones del resto de módulos desarrollados, siendo primero el de los guardias y

continuando con el de monitoreo y gerencia. Al momento de instalación de los dispositivos de hardware, fue necesario integrarlos en el sistema mediante la asociación de cada uno de ellos a la zona restringida sobre la cual opera. Gracias a esto, los distintos componentes del sistema son capaces de comunicarse con los dispositivos a través de MQTT. La configuración de un nuevo dispositivo requiere un nombre del master, una zona, un conjunto de puestos (slaves) sobre los que tiene control.

Para capacitar al personal en el uso del sistema, se realizaron demostraciones en los distintos puestos de trabajo sobre las funcionalidades destinadas a cada tipo de usuario. Esta actividad se llevó a cabo en reiteradas oportunidades para cubrir a todo el personal y todas sus dudas. Además, se brindaron dos presentaciones a distintos grupos de personal donde se expuso todo el flujo de trabajo en una proyección y, al finalizar, se evacuaron las dudas que presentaban.

Migración de datos

Al no contar con documentación o apoyo de algún responsable del sistema previo, fue necesario realizar un trabajo de ingeniería inversa sobre las bases de datos originales para poder obtener los datos y relaciones necesarias a migrar. Esta base de datos estaba desarrollada sobre el motor Microsoft SQL Server, por lo que fue necesaria la creación de un script para trasladar dichos datos a nuestro sistema de base de datos MySQL.

En un primer momento, hubo que hacer un reconocimiento de las tablas y columnas presentes en el sistema previo, ya que no se contaba con la misma estructura de datos que tenía el nuevo sistema. Así mismo, no necesariamente las tablas se correspondían entre sí, debido a que datos que antes se encontraban en una tabla, ahora se encontraban en dos o más tablas.

Un desafío importante fue la pobre normalización de la base de datos, por lo que hubo que realizar un trabajo extra para integrar datos y, así, adaptarlo a nuestro modelo. Unos de los principales inconvenientes en la estructura de datos de la base original era la replicación de información. Esto se hizo notable principalmente en la cantidad de información de trabajadores repetida. Por ejemplo, una persona que

trabajaba para tres diferentes empresas estaba cargada tres veces en el sistema. Para resolver este inconveniente, se agrupó toda esta información repetida en una única instancia, la cual fue luego relacionada al resto de entidades correspondientes.

Otro reto surgió al momento de ver que, debido a que el sistema previo no contaba con validaciones de datos, muchas veces en los campos se contaba con información distinta a la que era destinado originalmente. Esto implicaba que cuando se cargaran en el nuevo sistema las validaciones iban a fallar. Por ello, fue necesario adaptar algunos datos y descartar otros.

11. Consideraciones de licencia e innovación

En lo referido a la *innovación comercial*, se desarrolló un producto propio de la empresa, a diferencia de las principales soluciones disponibles en el mercado que son ofrecidas mediante un contrato de licencia tipo "alquiler". Además, la posibilidad de cambiar el fabricante de hardware, les permite poder elegir al que, en tiempos o precios, mejor los satisfaga.

En cuanto a la *innovación tecnológica* respecto a sistemas comerciales de similar funcionalidad existentes, se destaca la forma de comunicación entre las distintas componentes del sistemas. En nuestro caso, se permite comunicar dispositivos de hardware con un cliente web de manera directa, sin la necesidad de instalar los controladores de cada nuevo dispositivo que se agrega al conjunto. Para lograr esto se utilizó MQTT, una tecnología emergente usada en IoT. Otro ejemplo de la utilización del servicio sin necesidad de una instalación de drivers es el uso del protocolo AirScan. Este logra que un programa pueda comunicarse directamente a un scanner de red sin utilización de drivers, posibilitando el desarrollo de una solución distribuida que permite al usuario controlar cualquier escáner de red desde un navegador web. Además, la utilización del protocolo IPP, a través de CUPS, permite centralizar la impresión de credenciales logrando que el usuario use cualquier impresora autorizada.

Con la finalidad de crear un sistema web rico y moderno, se utilizaron algunas de las tecnologías más extendidas en la actualidad. La aplicación cumple con los estándares web que le permite funcionar en los navegadores modernos, es posible crear un acceso en la pantalla de inicio de cualquier dispositivo, se puede compartir fácilmente a través de una URL y puede adaptarse a distintos tamaños de pantalla, aunque no está optimizado para celulares ya que no fue un requerimiento. Por ello, el sistema es una aproximación a una PWA (del inglés *Progressive Web App*), ya que, con una configuración mínima, brinda al usuario una experiencia a la par de las características de una aplicación de escritorio. Sin embargo, no puede considerarse como una PWA propiamente dicha ya que no puede funcionar si el navegador no se encuentra conectado a la red local, no se encuentra indexada en los principales buscadores y no cuenta con un manifesto JSON.

Gracias a la realización de una aplicación SPA (del inglés *Single Page Application*), los tiempos de respuesta de la interfaz de usuario son mucho menores a los de una web convencional, debido a que se recarga la información deseada y no la página completa. Estas tecnologías brindan mayor seguridad y validación sobre los datos ingresados al sistema, así como la integridad de los mismos.

Finalmente, en cuanto a la estructura de datos se utilizó una base de datos SQL. La misma además hace utilización de columnas JSON, lo que implica la posibilidad de persistencia de datos de distinta naturaleza, dando cierta flexibilidad propia de una base de datos NoSQL. Por ello, se interpreta como una base de datos mixta.

12. Métricas

Métricas del proyecto

Al inicio del proyecto se realizó una planificación en la cual se establecía un estimado de tiempo por fase (análisis y diseño, desarrollo, implementación y documentación) y a su vez dentro de la fase de desarrollo por cada uno de los cuatro módulos (administración, guardias, monitoreo y gerencia). De acuerdo a una planificación estimada por funcionalidad, el análisis arrojó un total de *800 horas*.

Haciendo un estudio de los tiempos reales del proyecto se obtuvo que el tiempo total fue de *880 horas* para el análisis, diseño, desarrollo e implementación, y *100 horas* para la escritura del presente Informe Final. Siendo un total de *980 horas*, por lo que se observa una excedente, en cuanto a la planificación, de *180 horas*.

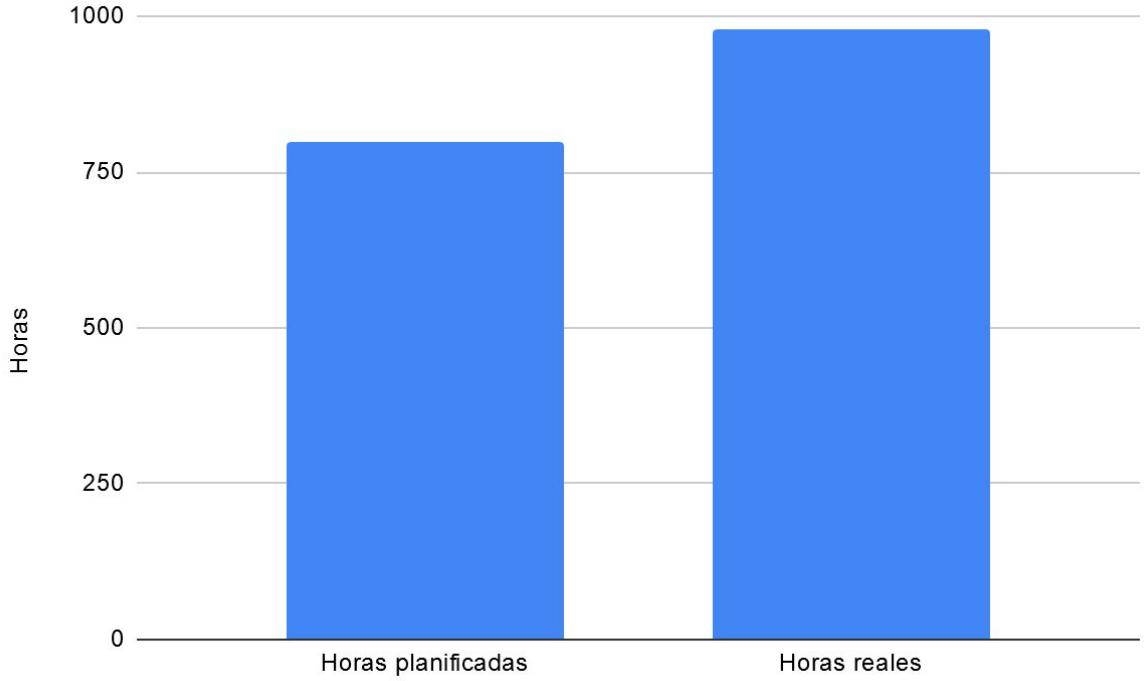


Gráfico 12.1 - Horas planificadas vs. horas reales

Si bien al inicio del proyecto se relevaron requerimientos y se hizo un análisis general del mismo, el ciclo de vida fue iterativo e incremental, por lo cual las fases de análisis, diseño, desarrollo e implementación no pueden ser discriminadas entre sí.

Dado a que una vez instalado un módulo se seguían recibiendo requerimientos o reporte de errores sobre el mismo por parte de los usuarios, se continuaba trabajando sobre ellos. Aquí se muestran la cantidad de horas invertidas por módulo:

Módulo	**Tiempo en horas**
Administración	420
Guardias	260
Monitoreo	140
Gerencia	60
Total	**880**

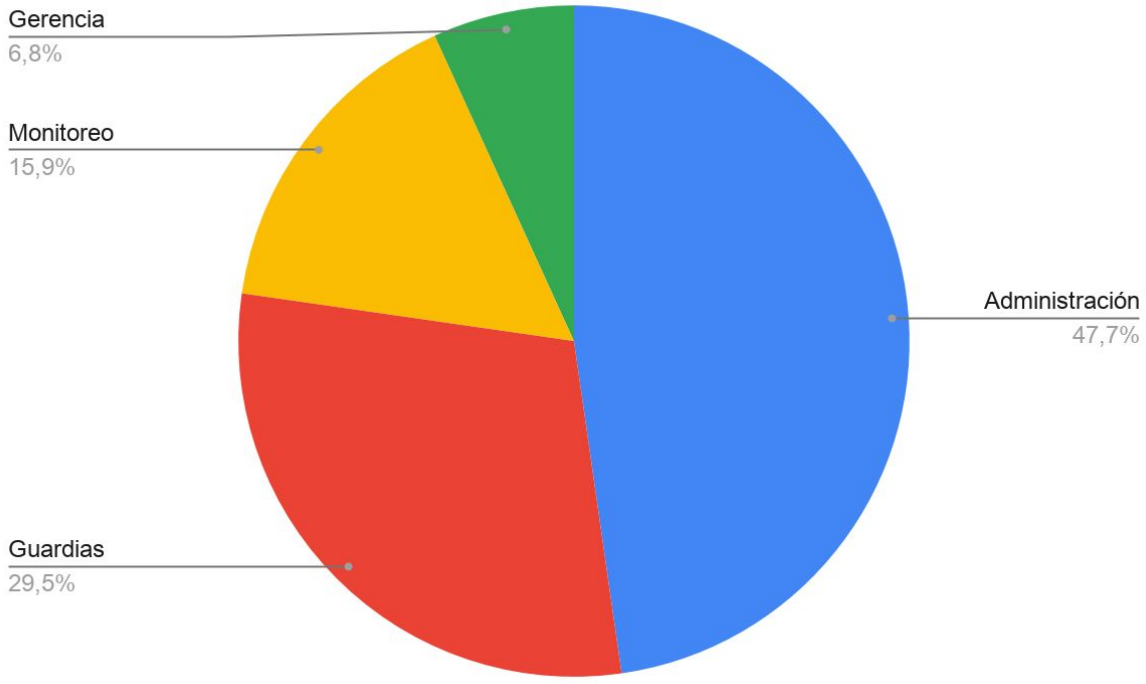


Gráfico 12.2 - Porcentaje de tiempo por módulo

Laravel, el framework elegido, utiliza PHP para el back-end, es aquí donde se encuentra la mayor cantidad de código, como se muestra en el *Gráfico 12.3*. Luego casi dos quintos del código son destinados a la interfaz de usuario, con la utilización de

Vue.js para el front-end. Finalmente, Python y Shell Script se utilizaron para comunicación con dispositivos de hardware.

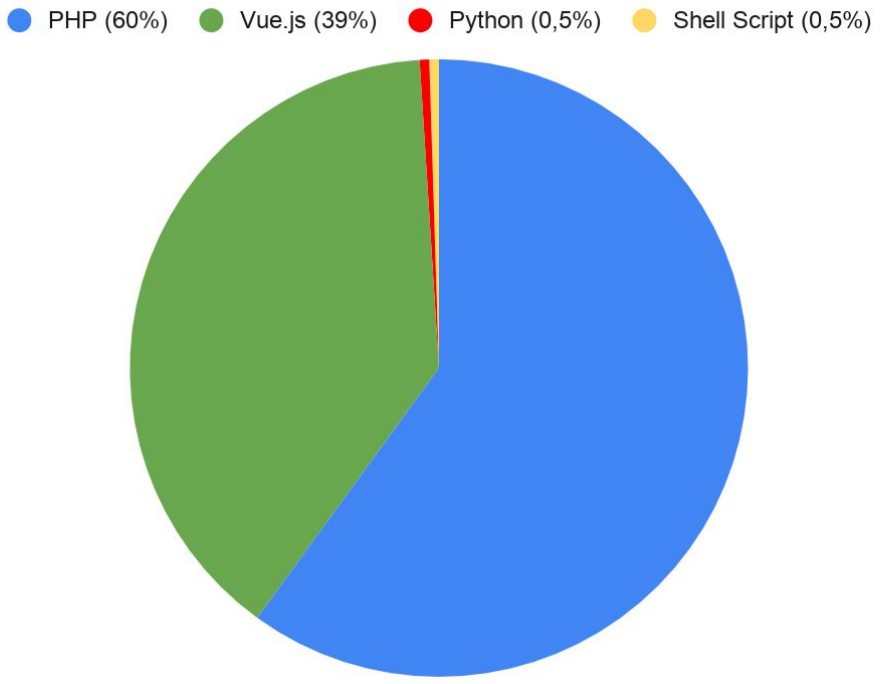


Gráfico 12.3 - Distribución de lenguajes de programación del proyecto

Métricas del sistema

Haciendo un estudio con *ApacheTop*, una herramienta de monitoreo de tráfico de peticiones HTTP, durante 5 minutos se observan 1.3 peticiones/segundo con un promedio de 177.1 KB/segundo. Estos valores son analizados en un horario normal, no en horario pico.

```
last hit: 14:32:40         atop runtime:  0 days, 00:05:11             14:32:41
All:          390 reqs (   1.3/sec)         53.1M (  177.1K/sec)     139.4K/req
2xx:      339 (86.9%) 3xx:      51 (13.1%) 4xx:      0 ( 0.0%) 5xx:      0 ( 0.0%)
R ( 30s):      40 reqs (   1.3/sec)       5799.6K (  193.3K/sec)     145.0K/req
2xx:       35 (87.5%) 3xx:       5 (12.5%) 4xx:      0 ( 0.0%) 5xx:      0 ( 0.0%)
```

Figura 12.1 - Captura del análisis de ApacheTop

Utilizando el sistema de bitácoras incorporado en el broker MQTT Mosquitto, teniendo en todo momento 11 dispositivos de hardware conectados, se obtuvo un

promedio de 66.3 mensajes enviados por minuto y un promedio de 68.45 mensajes recibidos por minuto.

Al hacer un análisis de la cantidad de consultas que realiza el sistema a la base de datos se obtienen los siguientes gráficos con la herramienta *Cacti*:

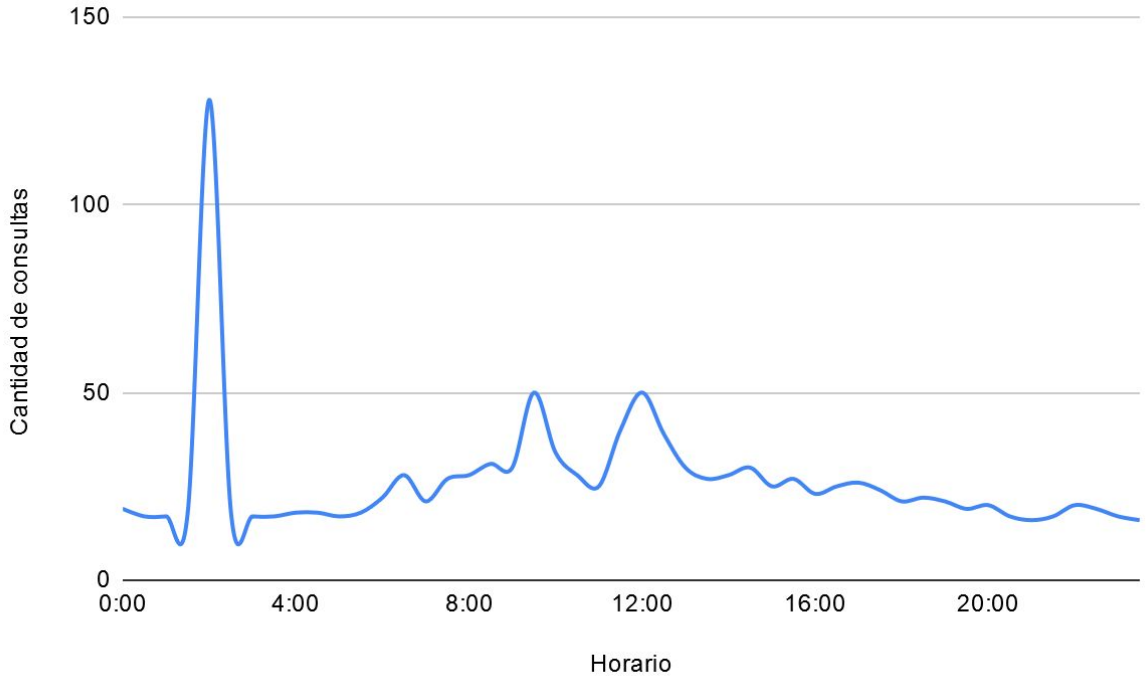


Gráfico 12.4 - Cantidad de consultas SQL diarias

En el *Gráfico 12.4* se observa cómo se distribuyen las consultas a lo largo de las 24 horas de un día semanal, pudiéndose observar en el mismo una serie de picos. El primero de ellos el que ocurre a las 2:00 AM, que corresponde al proceso de sincronización de la base de datos del servidor de aplicación con las bases de datos internas de las tarjetas de los distintos dispositivos de hardware. El resto de picos se observan cerca del mediodía, que es el momento donde más personas acceden y egresan de las zonas restringidas del puerto, y donde la oficina de administración recibe más personas para realizar trámites.

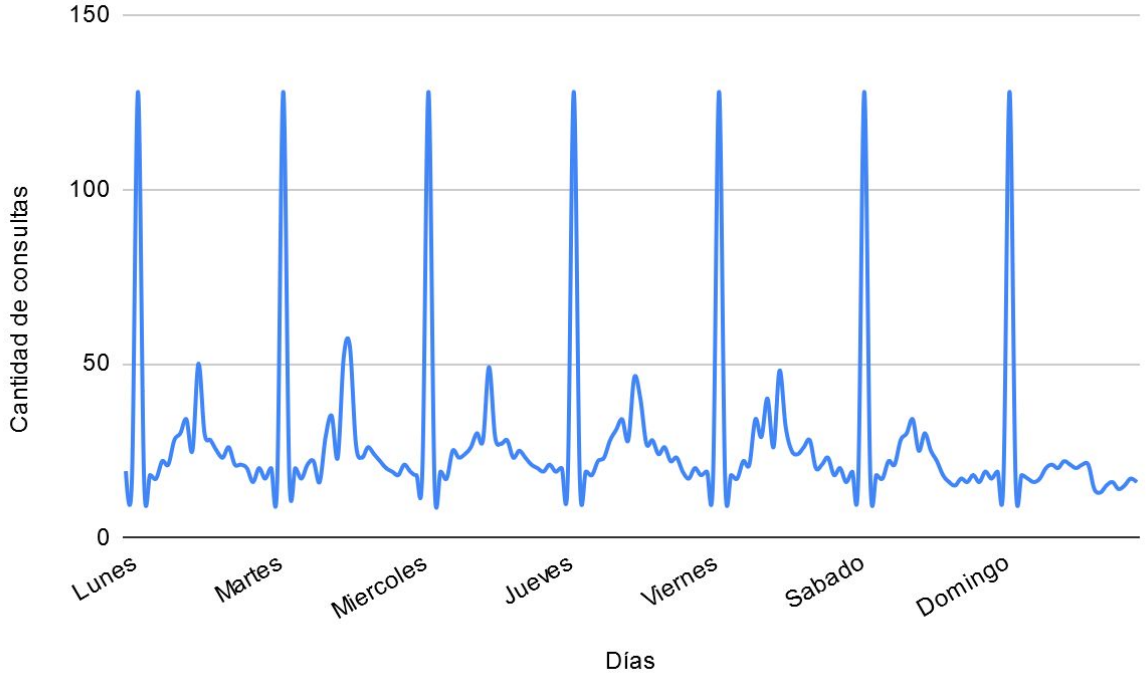


Gráfico 12.5 - Cantidad de consultas SQL semanales

En el *Gráfico 12.5* se observa la distribución de las consultas a lo largo de una semana. En los datos correspondientes entre el día Lunes y Viernes, puede observarse un comportamiento similar al observado en el *Gráfico 12.4*. La diferencia se nota en los días Sábado y Domingo. En un principio en estos días, la oficina de administración se encuentra cerrada, reduciendo así una cantidad significante de consultas, viéndose reflejado esto en los picos cercanos al mediodía. Por otro lado, el fin de semana es cuando la mayor cantidad de empleados no concurre a su lugar de trabajo.

Como se ha mencionado el sistema informa el estado del trabajador y si tiene el acceso habilitado o no, pero es el guardia quien tiene la última palabra en decidir si accede o no a las zonas restringidas, dejando constancia en el sistema. Por ello se presentan cuatro casos:

- Cuando el sistema decía que tenía el acceso habilitado, y el guardia lo dejó ingresar.
- Cuando el sistema decía que tenía el acceso inhabilitado, y el guardia no lo dejó ingresar.

- Cuando el sistema decía que tenía el acceso habilitado, y el guardia no lo dejó pasar. Esta decisión va en contra del estado informado por el sistema.
- Cuando el sistema decía que tenía el acceso inhabilitado, y el guardia lo dejó pasar. Esta decisión va en contra del estado informado por el sistema.

Haciendo un estudio de las bitácoras de entradas, que al momento de realizar este análisis posee 1.968.431 accesos registrados, se obtiene la siguiente información:

Accede con acceso habilitado	94.98%
No accede con acceso inhabilitado	0.43%
Accede con acceso inhabilitado	4.58%
No accede con acceso habilitado	0.0063%

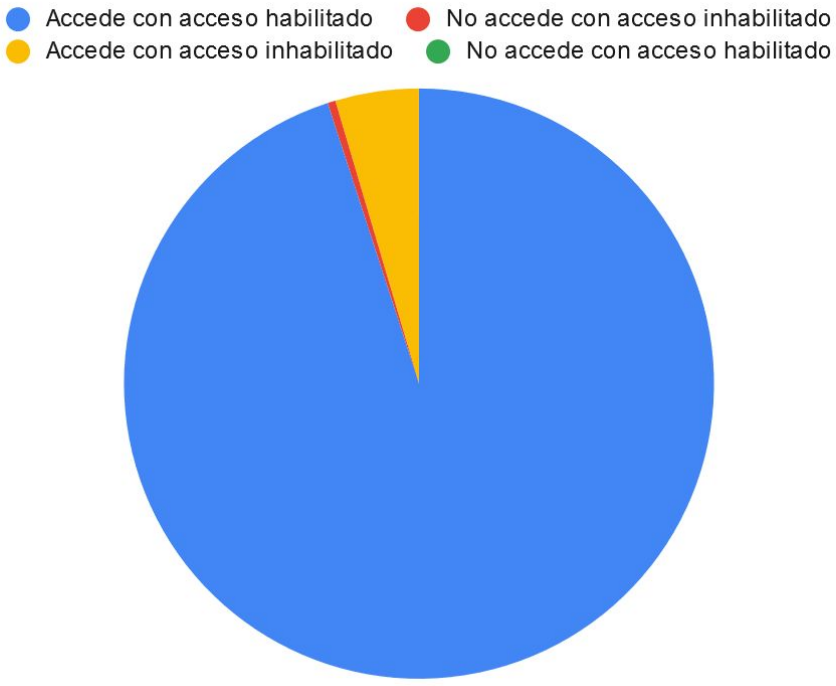


Gráfico 12.6 - Distribución de tipos de acceso

Estos datos informan que la gran mayoría de las veces el guardia toma la decisión en consonancia con el sistema, por lo que el sistema informa el estado correctamente.

13. Conclusiones

Para ambos autores del presente informe, este fue el primer trabajo real en lo referido a la Ingeniería Informática, lo cual nos puso al frente de muchas situaciones que hasta el momento eran un ideal.

Al momento de concluir el proyecto final se realiza una retrospección sobre los buenos aspectos que tuvo el proyecto, y aquellos que pudieron haber sido mejores.

Dentro de lo enriquecedor del proyecto se encuentran muchas lecciones aprendidas. La relación con el cliente fue muy importante, ya que tuvimos que aprender a interactuar con un cliente no especializado en temas informáticos, por lo que fue necesario conducirlo sobre un camino que permita obtener los requerimientos del proyecto. En cuanto a lo técnico, se afianzaron conocimientos sobre los lenguajes, frameworks y técnicas utilizadas. Y por otro lado, se adquirió mucho conocimiento sobre aspectos tecnológicos que no fueron abordados durante la carrera y que son muy enriquecedores, como es el caso de la comunicación entre los distintos dispositivos del sistema (MQTT).

Muchos de los conocimientos adquiridos de manera teórica durante la carrera toman mayor relevancia al ser utilizados en la práctica en un proyecto real. Algunos ejemplos de esto son los distintos tipos de sistemas distribuidos, patrones de diseño de software (Model View Controller, Observer-Observable), redes y aspectos de seguridad informática.

Al ser los únicos desarrolladores del sistema y trabajar de manera autónoma, las decisiones eran tomadas por nosotros, siempre con una supervisión. De esta manera, éramos nosotros quienes estábamos buscando soluciones e investigando sobre las distintas tecnologías utilizadas, y así desarrollando un lado autodidacta que en el futuro va a ser un gran complemento.

El hecho de ser nuestro primer trabajo también trae una serie de puntos negativos que pueden ser mejorados en un futuro con la experiencia adquirida. Si bien el desvío porcentual entre el tiempo planificado y el real fue mínimo, se considera como

un aspecto a fortalecer en proyectos futuros. Algunos eventos que impactaron en la duración del desarrollo fueron cambios imprevistos en los requerimientos, etapas que requirieron más tiempo de lo planificado y atraso del cliente en cuanto a la compra de los insumos requeridos.

Al querer escuchar al gran volumen de usuarios de manera individual, y querer satisfacerlos a todos en cuanto a sus requerimientos, observamos que muchos de ellos no tenían los mismos intereses para con el sistema. Esto también es debido a que no tienen un flujo de trabajo estandarizado. Este punto fue lo que más hizo que nos atrasemos, ya que se tuvieron que modificar cosas ya desarrolladas o agregar cosas nunca especificadas. Esto se podría evitar definiendo con el cliente un referente funcional que conozca el flujo de trabajo en su totalidad.

La división de tareas fue otro aspecto influyente en el retraso del proyecto. En un principio, ambos nos encontrábamos realizando las mismas tareas de manera conjunta cuando se podrían haber dividido las mismas para agilizar el desarrollo. Esto fue mejorando a lo largo del proyecto, por lo que puede considerarse como una habilidad adquirida.

Todos estos aspectos que de un modo son negativos, porque retrasaron lo que fue la estimación del tiempo del proyecto, fueron aspectos positivos en lo personal, debido a que son lecciones aprendidas que le permiten a uno tomar mejores decisiones a futuro.

En cuanto al sistema en sí, luego de un tiempo considerable de encontrarse en funcionamiento, se puede decir que cumple su función de manera correcta, más allá de algunos aspectos que puedan ser mejorados.

La escritura del informe se vio demorada debido a que durante el desarrollo del sistema seguíamos con las cursadas de la carrera, exámenes, y otros trabajos, lo cual no nos permitió realizar previamente el informe en paralelo con el sistema. Pero, por otro lado, como ya se mencionó el sistema tuvo muchas modificaciones a lo largo del proyecto, por lo que el informe fue escrito una vez ya estable el mismo. De esto modo, se pudieron profundizar los temas con exactitud. Además, el sistema al tener ya un tiempo

de funcionamiento nos permitió poder generar estadísticas y estudios con una gran bitácora de datos.

Para concluir, se puede afirmar que con el desarrollo del proyecto adquirimos competencias ingenieriles referidas a la Informática, como:

- Identificar, formular y resolver problemas de ingeniería.
- Concebir, diseñar y desarrollar proyectos de ingeniería.
- Gestionar -planificar, ejecutar y controlar- proyectos de ingeniería
- Usar de manera eficaz las técnicas y herramientas de la ingeniería.

Este ha sido un gran proyecto, en cuanto a lo personal, para ser nuestro primer trabajo en la materia de Informática, debido a la gran magnitud del mismo y las tecnologías utilizadas. Además, es un sistema desarrollado para el CPRMDP, la administración de un sector emblemático de la ciudad en la que crecimos, generandonos mucho orgullo.

14. Trabajos Futuros

El sistema puede presentar una gran escalabilidad. Hay proyectos o mejoras que se pueden realizar sobre el mismo a futuro. Uno de ellos, gracias a que los clientes del sistema solo requieren de un navegador web para funcionar, es la adaptación de las interfaces de usuario a dispositivos móviles, para poder utilizar el sistema desde una tableta, por ejemplo. Este requerimiento no fue solicitado en un principio ya que solo se utiliza en computadoras de escritorio.

Otra mejora es brindar la posibilidad, a los trabajadores y/o empresas que acceden a las zonas restringidas, de consultar fechas de vencimiento y documentación faltante a través de una página web accesible por Internet desde cualquier lugar, así como actualizarla de manera remota. Esto disminuiría la cantidad de gente que tenga que presentarse en las oficinas del consorcio, facilitando el trabajo a sus empleados. Otro beneficio sería la reducción en el uso de papel al no tener que presentar la copia física de la documentación.

Para el control de anomalías, se propone un estudio sobre el histórico de ingresos y egresos, con el fin de crear reglas de inferencia que permitan identificar acciones sospechosas por parte de los trabajadores. Por ejemplo, dos ingresos consecutivos sin una salida intermedia. Al día de la fecha, el sistema cuenta con más de un millón de accesos y egresos registrados, lo cual es una gran base de información que puede ser utilizada para entrenar una red neuronal capaz de detectar anomalías en nuevas entradas que sucedan en el futuro. Gracias a esta red neuronal, se pueden desarrollar funcionalidades basadas en bots, complementarias al sistema, para colaborar y asistir en la función de los guardias. Habría que tener consideración especial con los trabajadores que se embarcan, ya que estos pueden embarcarse, egresar del puerto sin pasar por una de las terminales controladas por el sistema, y luego volver a la ciudad por tierra para ingresar nuevamente por una terminal.

Para solventar este último inconveniente, contando con la información de la prefectura naval acerca de las distintas embarcaciones que se encuentran dentro del puerto de la ciudad en un momento dado y quienes son aquellas personas que se

embarcan, es posible cruzar estos datos para así no considerarlos como una anomalía en las reglas de inferencia, sino como un caso eventual en el control de acceso.

Finalmente, el sistema de control de acceso aquí mencionado está orientado al acceso portuario, ya que es para quien fue destinado. Pero al abstraer las generalidades con cualquier otro ámbito, y realizando algunas modificaciones para adaptarlo, se puede llevar a una solución aplicable en otro contexto.

15. Glosario

Acceso: acción de ingresar o egresar del puerto por una de sus zonas seguras. Requiere una tarjeta activa y poseer la documentación actualizada.

Cloud Computing: disponibilidad de recursos informáticos a través de una red, especialmente almacenamiento y capacidad de cómputo, sin una gestión activa por parte del usuario. El término generalmente es usado para describir los centros de datos disponibles desde cualquier lugar para los usuarios a través de Internet desde cualquier dispositivo.

Consorcio Portuario Regional de Mar del Plata (CPRMDP): cliente que solicita el sistema de acceso.

FTP: protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

Guardia: personal del CPRMDP encargado de controlar el acceso de las distintas personas a través de los ingresos a las zonas seguras del puerto.

HTTP: protocolo de la capa de aplicación para la transmisión de documentos hipermedia. Fue diseñado para la comunicación entre los navegadores y servidores web. Sigue el clásico modelo cliente-servidor. Se trata de un protocolo sin estado, lo que significa que el servidor no guarda ningún dato entre dos peticiones.

IoT: del inglés *Internet of Things*, en español *Internet de las Cosas*. Sistema interrelacionado de dispositivos de computación y equipos mecánicos o digitales, cada uno con un identificador único y la habilidad de transferir información a través de una red sin requerir de interacción humano-humano o humano-computadora.

JSON: formato de archivo standard para el intercambio de información. Utiliza texto plano para almacenar y transmitir datos, usando objetos que consisten en pares clave-valor. Deriva de JavaScript, pero en la actualidad es independiente del lenguaje de

programación, ya que todos los lenguajes modernos poseen una forma de crear y leer JSON.

Log: documentación producida automáticamente y con una marca temporal de eventos relevantes a un sistema en particular.

Machine Learning: subcampo de las ciencias de computación y una rama de la inteligencia artificial dedicada al estudio de algoritmos computacionales que mejoran automáticamente a través de la experiencia.

Nivel de riesgo: determinación de acceso de personal según el contexto de seguridad. Hay tres niveles, cuanto mayor el nivel mayor la restricción para el ingreso.

Nube: red enorme de servidores remotos de todo el mundo que están conectados para funcionar como un único ecosistema. Estos servidores están diseñados para almacenar y administrar datos, ejecutar aplicaciones o entregar contenido o servicios, como streaming de vídeos, correo web, software de ofimática o medios sociales.

ORM: técnica de programación que consiste en la transformación de las tablas de una base de datos en una serie de entidades utilizando un lenguaje de programación orientado a objetos. Esto crea, en efecto, una base de datos virtual que puede ser utilizada dentro del lenguaje de programación para facilitar las consultas a la base de datos.

PIN: número identificatorio de la tarjeta de ingreso del trabajador. Es único por cada tarjeta generada.

Red neuronal: red interconectada masivamente de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.

Trabajador: persona ajena al CPRMDP que debe ingresar a una o varias zonas restringidas del puerto para ejercer sus actividades laborales.

16. Bibliografía

1. Van Steen M. y Tanenbaum A.S. (2017). *Sistemas Distribuidos*. Tercera edición.
2. Coulouris G., Dollimore J. y Kindberg T. (2001). *Sistemas Distribuidos, conceptos y diseño*. Pearson Educación S.A.
3. Larman C. (2001). *UML y patrones. Introducción al análisis y diseño orientado a objetos.* Segunda Edición. Pearson Educación S.A.
4. Sanchez J.R., Garcia Bernal R, Manosalva Osorio, Rezende Sydney y Sgut M. (2004). *Protección Marítima y Portuario en América del Sur. Implementación de las medidas y estimación de gastos*. IIRSA-CEPAL. Santiago de Chile.
5. Mundo Maritimo Ltda, Pejovés J.A. (2019). *El concepto de puerto seguro en los contratos de fletamento*. Recuperado el 10/07/2020 de https://www.mundomaritimo.cl/noticias/el-concepto-de-puerto-seguro-en-los-contratos-de-fletamento
6. Oracle Corporation (2020). ¿Qué es Internet of Things?. Recuperado el 19/08/2020 de https://www.oracle.com/ar/internet-of-things/what-is-iot.html
7. Taylor Otwell (2011-2020). *Laravel - The PHP Framework for Web Artisans*. Recuperado el 10/07/2020 de https://laravel.com/docs/5.6
8. Evan You (2014-2020). *Vue.js - The Progressive JavaScript Framework*. Recuperado el 10/07/2020 de https://vuejs.org/v2/guide/
9. The PHP Group (2001-2020). *PHP Hypertext Preprocessor*. Recupeado el 10/07/2020 de https://www.php.net/
10. Oracle Corporation (2020). *MySQL*. Recuperado el 10/07/2020 de https://www.mysql.com/
11. Eclipse Mosquitto. *Mosquitto - An open source MQTT broker*. Recuperado el 10/07/2020 de https://mosquitto.org/

12. Mozilla (2005-2020). *MDN Web Docs*. Recuperado el 10/07/2020 de https://developer.mozilla.org/

Anexo I

Utilización del sistema

Como fue mencionado hay cuatro módulos para los distintos tipos de usuarios que presenta el sistema.

Cuando un usuario del **módulo de administración** ingresa al sistema de encuentra con la siguiente vista:

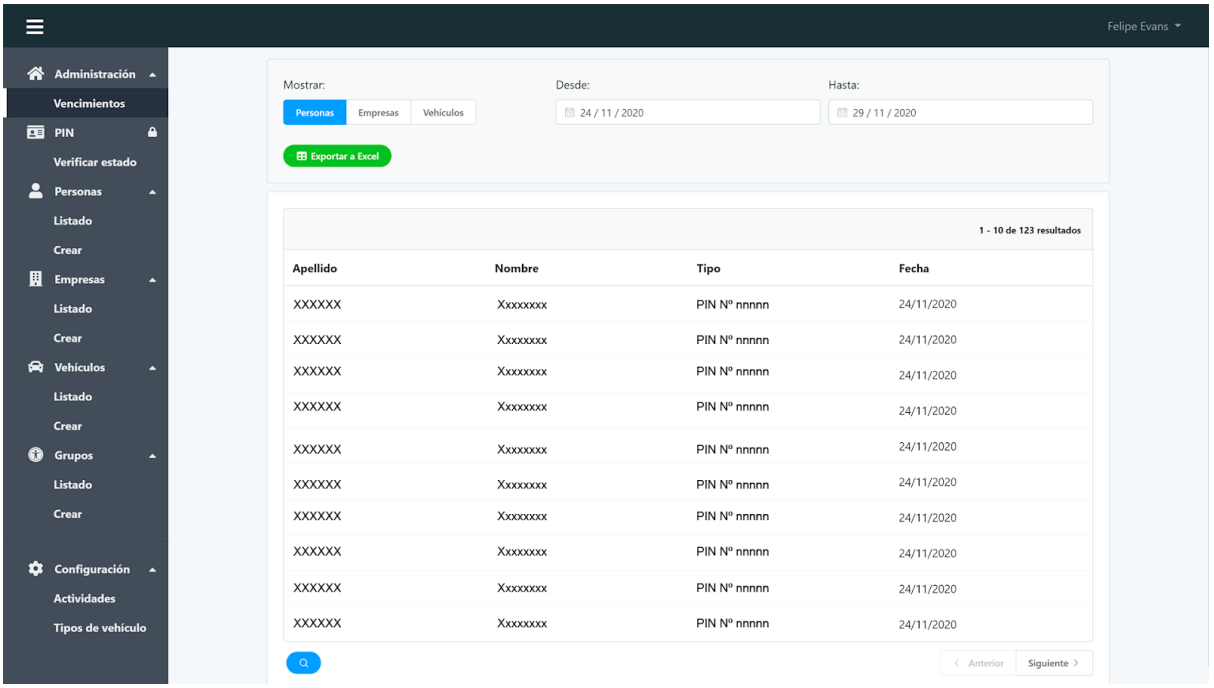


Figura 17.1 - Listado de vencimientos

La vista presenta los vencimientos según los filtros aplicados. Pueden discriminarse los listados entre las entidades de Personas, Empresas o Vehículos. También puede exportarse el listado para realizar informes o comunicaciones internas.

En la barra lateral se observan todos las entidades del sistema, así como los botones para visualizar el índice o para realizar una nueva creación.

Para la creación de una Persona se tienen cuatro pestañas de carga: información personal, información laboral, asignación de vehículos y documentación.

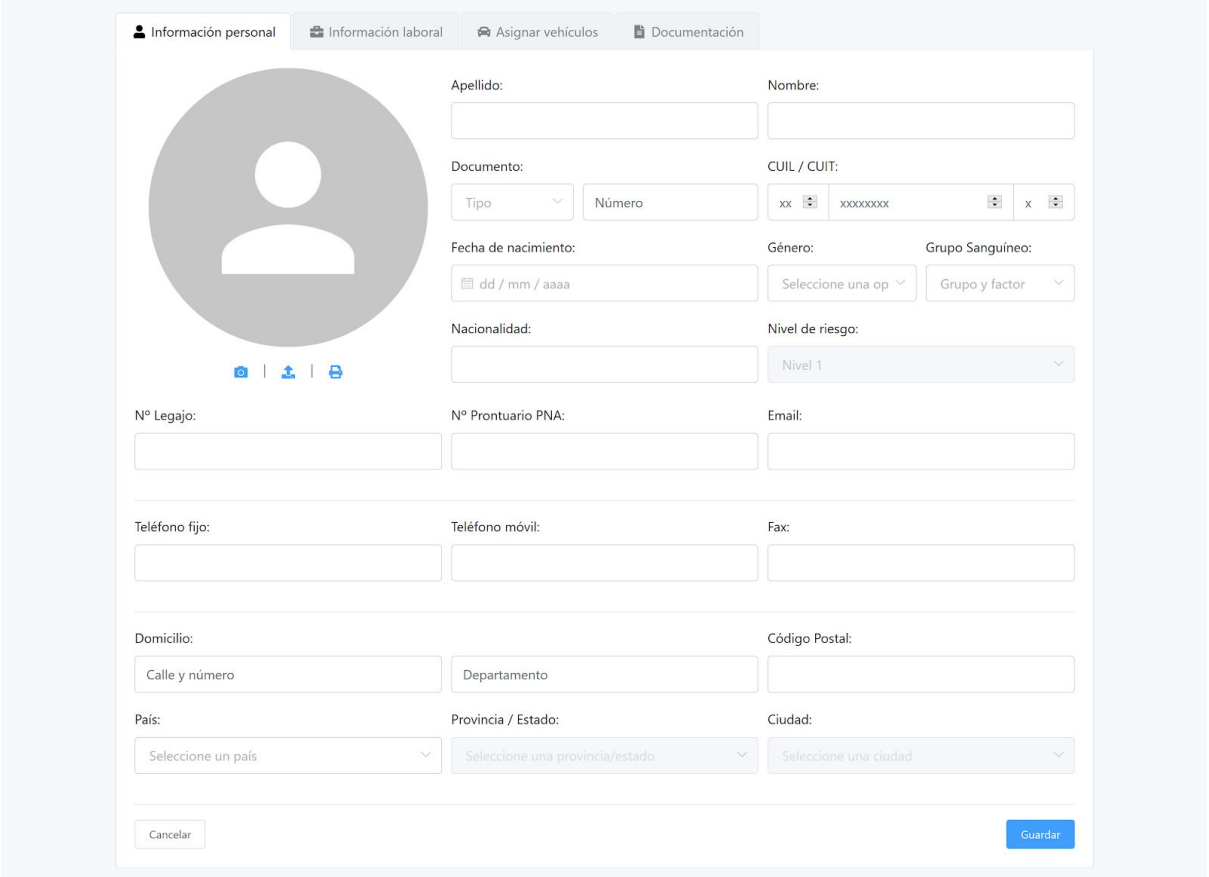


Figura 17.2 - Pestaña de información personal en creación de persona

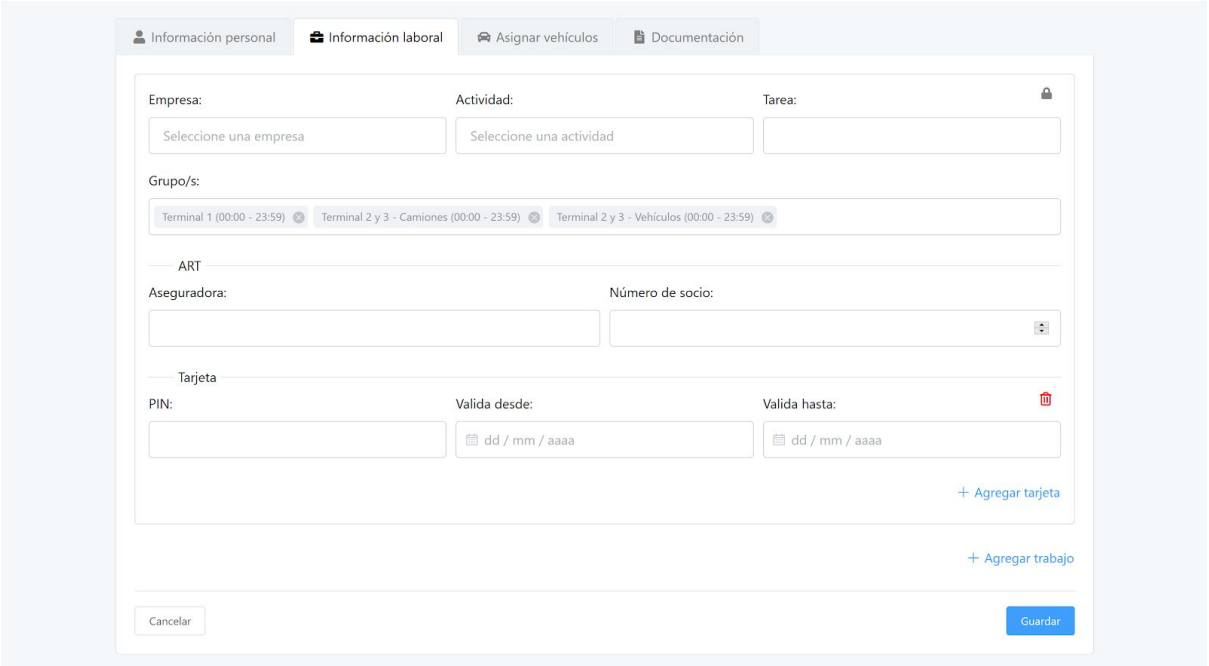


Figura 17.3 - Pestaña de información laboral en creación de persona

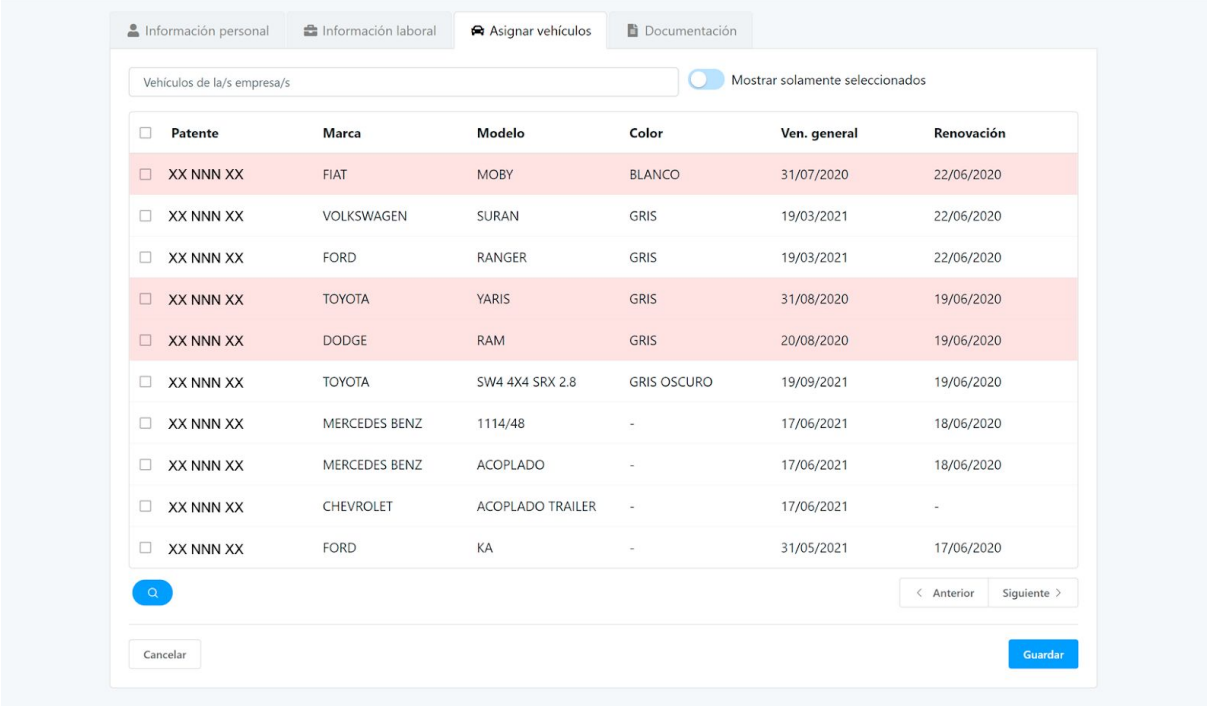


Figura 17.4 - Pestaña de asignación de vehículos en creación de persona

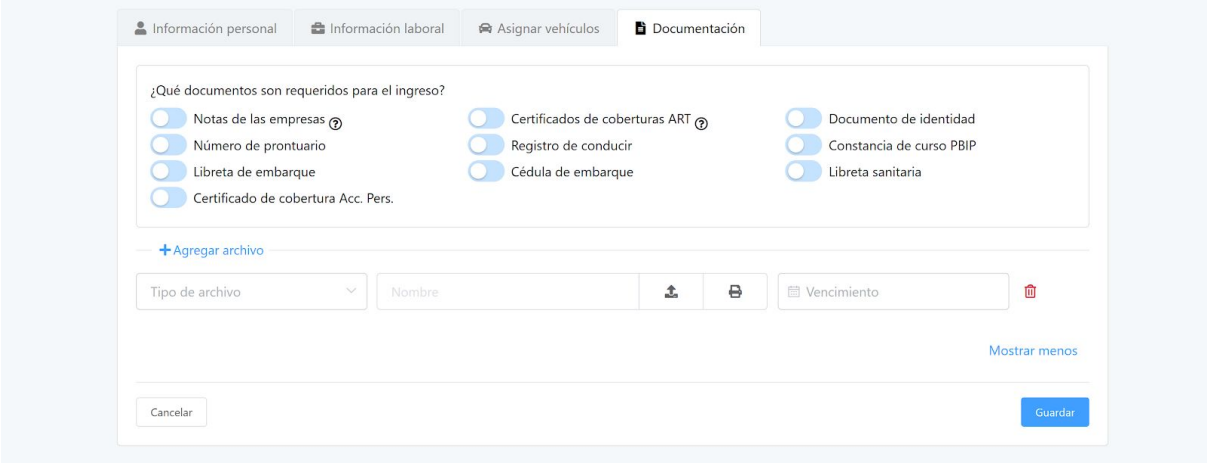


Figura 17.5 - Pestaña de documentación en creación de persona

La primer pestaña (*Figura 17.2*) de la creación de la persona está destinada a carga de datos personales así como una foto de perfil utilizada para imprimir en la tarjeta de acceso. En la segunda pestaña (*Figura 17.3*) se cargan los datos referidos al trabajo. Aquí se ingresa la empresa, la actividad que realiza dentro de ella, los accesos que tiene habilitados y, también, la información del PIN. En la pestaña de asignación de vehículos (*Figura 17.4*) se muestra el listado de vehículos del sistema, así como posibilidades de filtrar el mismo, para seleccionar cuales se le van a asignar a la persona creada. Los vehículos que están marcados en rojo son aquellos que fueron

deshabilitados. Finalmente, en la última pestaña (*Figura 17.5*) se carga la documentación. Además, en la parte superior se muestra *switches* sobre todos los ítems de documentación que pueden ingresar. De ser marcado algún ítem, el vencimiento que tenga ese documento será contemplado al momento del ingreso.

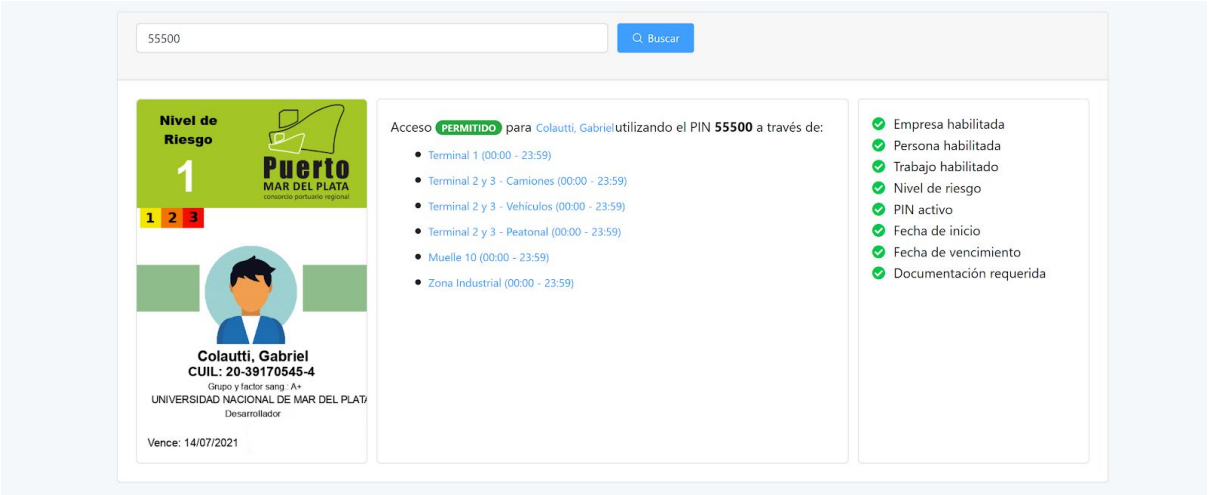


Figura 17.6 - Checkeo de estado de PIN

Para el ingreso de un trabajador son muchas las corroboraciones que realiza el sistema para habilitar o no a acceder, pero no son los guardias los encargados de ver cuál es el motivo por el que el trabajador pueda no tener el acceso habilitado. Por ello a los usuarios de administración se les ofrece esta vista (*Figura 17.6*) donde se le detallan todos los ítems que son contemplados en el checkeo del estado del PIN.

En el **módulo de guardias**, al iniciar sesión tienen el *libro del guardia* y unos buscadores en caso que deseen buscar al trabajador de forma manual.

En caso que el trabajador pase su tarjeta por el lector, o que el guardia busque por PIN o DNI de forma manual, se les abre una pestaña como se muestra en la *Figura 17.7*.

Aquí se muestra la información del trabajador relevante para el guardia al momento del ingreso. A la izquierda, se presenta la última copia de la credencial con información del trabajador y si está permitido o no el acceso según la información del sistema.

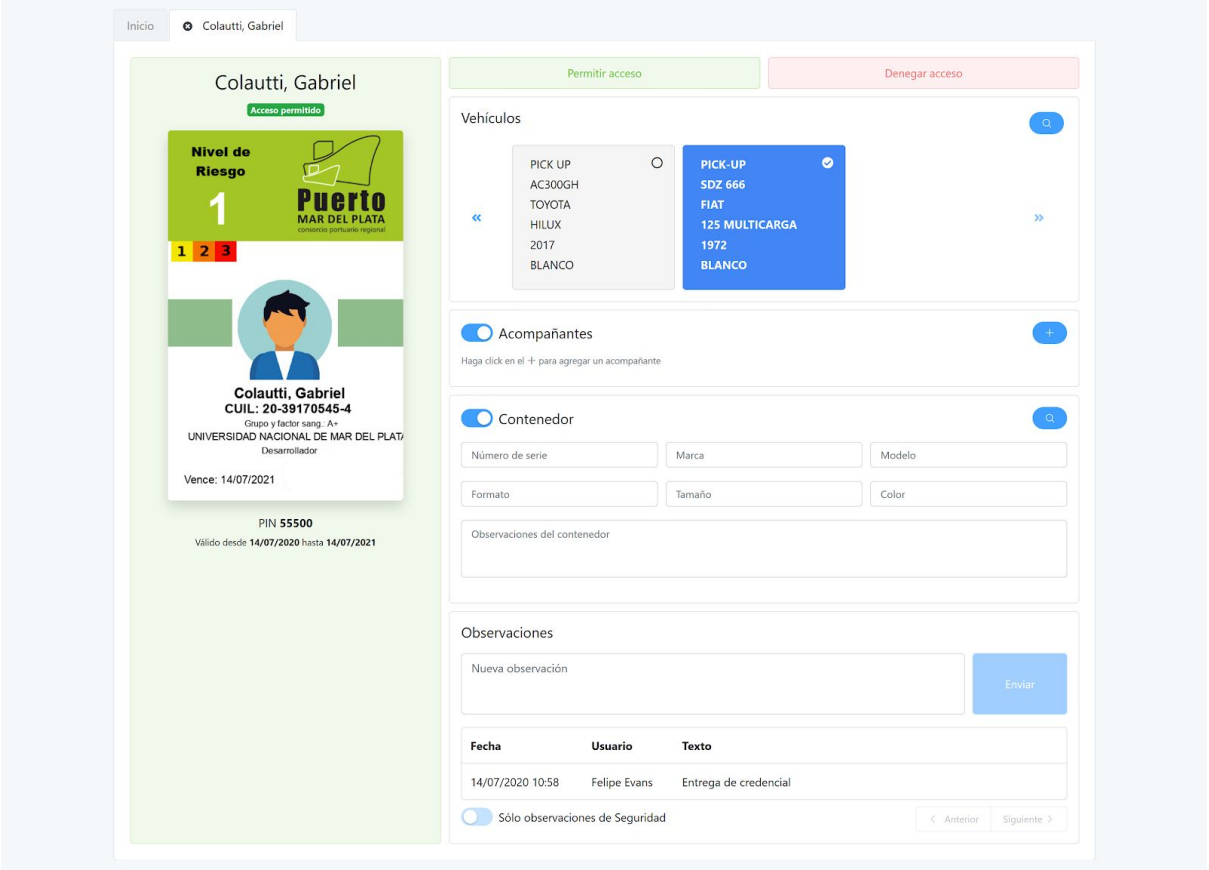


Figura 17.7 - Información del trabajador al ingreso

En el cuadro de la derecha se presenta la información, a completar por el guardia, referida al ingreso. En esta sección se ingresa el vehículo con el que ingresó (en caso que lo haya hecho con uno), los acompañantes que ingresaron con el conductor en el vehículo, si es un vehículo de carga e ingresa con un contenedor se debe ingresar la información del mismo, y, finalmente, puede ingresar observaciones que van a ser visibles por todos los usuarios del sistema.

Luego de completar todo, el guardia tiene la decisión final sobre si permitir o no el ingreso apretando sobre los botones en la parte superior. Si la decisión es contraria a la del sistema, se abrirá un cuadro de diálogo para ingresar el motivo de la decisión.

El usuario del **módulo de monitoreo** tiene la misma vista que el usuario de administración, pero no tiene permisos de escritura, por lo que no puede crear ni editar ninguna entidad.

Además de lo mencionado, estos usuarios tienen acceso al histórico de entradas.

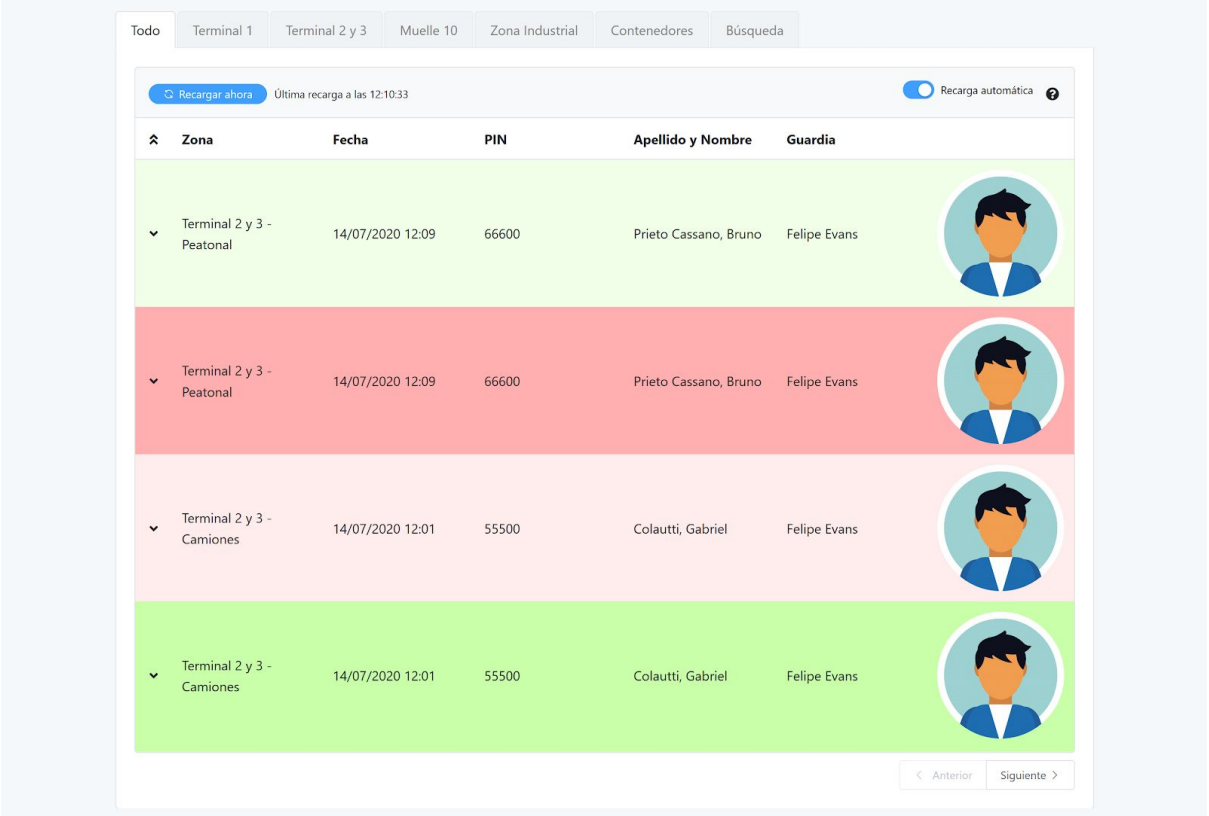


Figura 17.8 - Histórico de entradas

Aquí se observan las entradas con una información básica y el color indica el estado de cada una. Hay cuatro casos: ingresó y el sistema decía que estaba habilitado (verde fuerte), ingresó y el sistema decía que no estaba habilitado (verde tenue), no ingresó y el sistema decía que estaba habilitado (rojo tenue), y no ingresó cuando el sistema decía que no estaba habilitado (rojo fuerte).

También las entradas pueden ser filtradas por terminal, o realizar una búsqueda intensiva sobre el listado.

Finalmente, el **módulo de OPIP** cuenta con las mismas vistas y permisos del módulo de administración y el módulo de monitoreo combinados. Como funcionalidad extra, presenta la creación de usuarios y el cambio de nivel riesgo del puerto.

Eloquent ORM

Laravel incluye un ORM llamado Eloquent, el cual permite trabajar con el contenido y las relaciones de las base de datos usando una sintaxis expresiva. En Laravel, cada tabla de una base de datos corresponde a una clase de PHP llamada *Modelo*; para acceder a los datos de estas tablas se utilizan los métodos provistos por cada modelo.

Para ilustrar la simplicidad que provee utilizar este ORM, se presenta a continuación una serie de ejemplos de cómo se realizan ciertas operaciones utilizando Eloquent y su contraparte en SQL plano.

Suponiendo un modelo *Persona* que representa a la tabla *personas*, el código para seleccionar los últimos cinco elementos de dicha tabla es:

```
Persona::orderByDesc('id')
       ->limit(5)
       ->get();
```

```
select *
from `personas`
order by `id` desc
limit 5
```

Si no se desea seleccionar todos los datos de la tabla, se puede especificar aquellas columnas necesarias.

```
Persona::select('id', 'nombre')
       ->limit(5)
       ->get();
```

```
select `id`, `nombre`
from `personas`
limit 5
```

A la hora de seleccionar una persona determinada utilizando su clave primaria se utiliza el siguiente código:

```
Persona::find(20);
```

```
select *
from `personas`
where `id` = 20
limit 1
```

A la hora de trabajar con las relaciones entre distintas tablas, Eloquent provee un sistema de asociaciones y métodos que permite realizar dicha tarea de una manera

sencilla y elegante. Suponiendo que el modelo Persona posee una relación muchos a muchos con un modelo *Vehículo*, que representa la tabla *vehículos*, para obtener todos los vehículos de una persona determinada que sean de color rojo se utilizan las siguientes funciones:

```
Persona::find(20)
       ->vehiculos()
       ->where('color', 'rojo')
       ->get();
```

```
select *
from `vehiculos`
inner join `p_v`
on `p_v`.`veh_id` =
   `vehiculos`.`id`
where `p_v`.`per_id` = 20
and `color` = `rojo`;
```

Las relaciones se determinan en cada modelo. Si se sigue la convención de nombres del framework, el mismo es capaz de inferir el nombre de las tablas a utilizar en cada caso. Las clases que representan a Persona y Vehículo respectivamente son:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;

class Persona extends Model
{
      public function vehiculos()
      {
            return $this->belongsToMany('Vehiculo');
      }
}
```

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;

class Vehiculo extends Model
{
      public function personas()
      {
            return $this->belongsToMany('Persona');
      }
}
```

A la hora de crear o modificar las tablas, Eloquent provee un sistema de migraciones el cual permite generar el código SQL a través de funciones de php. Como ejemplo, se muestra el código utilizado para crear la tabla asociada al modelo Persona y el respectivo código SQL generado:

```
Schema::create('personas', function (Blueprint $table) {
     $table->bigIncrements('id');
     $table->string('apellido')->required();
     $table->string('nombre')->required();
     $table->string('cuil', 11)->unique();
     $table->timestamps();
});
```

```
CREATE TABLE `personas` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `apellido` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `nombre` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `cuil` varchar(11) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

En este anexo se han ejemplificado las funcionalidades básicas que otorga Eloquent. Si desea verse más ejemplos, funcionalidades avanzadas o entender mejor su funcionamiento en el conjunto del framework, referirse a la documentación oficial de Laravel.

Validación de datos

Laravel permite validar los datos enviados por los usuarios a través de los distintos formularios de la aplicación de manera sencilla. De esta manera se puede asegurar la consistencia e integridad de la información en las distintas tablas de la base de datos.

El principal método para validar la información enviada a través de una solicitud es el método *validate* de la clase *Request*, clase que engloba todo lo relacionado a la petición del usuario. Dicho método requiere como parámetro un array clave-valor, donde cada clave es un atributo a validar y el valor es otro array con las reglas de validación para dicho atributo. Si la validación es satisfactoria, el código sigue su ejecución normal, pero en caso de existir algún error en los datos enviados, el método arrojará una excepción con código de error 422 (código utilizado por Laravel para errores de validación) y un mensaje por cada atributo que sea inválido.

Por defecto, Laravel posee una amplia cantidad de validaciones disponibles, las cuales pueden ser encontradas en la documentación del framework. Si estas validaciones no fueran suficientes, es posible crear reglas personalizadas que se adapten a nuestras necesidades. Un ejemplo de esto es la regla de validación que verifica que el número de cuil sea válido, cumpliendo con la suma del código verificador:

```
// app/Providers/AppServiceProvider.php
Validator::extend('valid_cuil', function ($attribute, $val, $parameters)
{
    $regex = "/^(20|23|24|27|30|33|34)-[0-9]{8}-[0-9]{1}$/";
    if(!preg_match($regex, $val))
        return false;
    $sum = 0;
    $digits = str_split(str_replace('-', '', $val));
    $val_digit = array_pop($digits);
    for( $i = 0; $i < count($digits); $i++ )
        $sum += $digits[ 9 - $i ] * ( 2 + ($i % 6) );
    $val_number = 11 - ($sum % 11);
    $val_number = ($val_number === 11) ? 0 : $val_number;
    return $val_number == $val_digit;
});
```

Un ejemplo simple de validación para la información de una persona es el siguiente, donde el apellido y nombre deben estar presentes y tener como máximo 255 caracteres de longitud, y el cuil también debe estar presente y ser único en la tabla de personas:

```
public function store(Request $request)
{
    $request->validate([
        'apellido' => ['required', 'max:255'],
        'nombre' => ['required', 'max:255'],
        'cuil' => ['required', 'unique:personas', 'valid_cuil']
    ]);

    // La información de la persona es correcta...
}
```

Archivos de tarjetas para dispositivos de hardware

Como se ha mencionado en el informe, los dispositivos de hardware actualizan su base de datos interna. Siguiendo la estructura descrita, se presenta un archivo de ejemplo para un índice:

```
20071301.csv;03;17998;03;17998
20071302.csv;04;00000;04;02158
20071303.csv;04;02159;04;04073
20071304.csv;04;04074;04;06319
20071305.csv;04;06321;04;08703
20071306.csv;04;08705;04;11917
20071307.csv;04;11931;04;14723
20071308.csv;04;14734;04;17314
20071309.csv;04;17317;04;19447
20071310.csv;04;19448;04;21383
20071311.csv;04;21384;04;23065
20071312.csv;04;23070;04;24589
20071313.csv;04;24591;04;25520
20071314.csv;04;25526;04;26380
20071315.csv;04;26381;04;27121
20071316.csv;04;27122;04;27780
```

Por otra parte, se presenta como ejemplo el archivo 20071309.csv contenido en dicho índice:

```
04;17317;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17325;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17328;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17331;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17332;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17333;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17334;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17347;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17350;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
04;17353;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
                            ...
04;19447;0000;2359;xxxx;xxxx;1;1;1;1;1;1;%;1
```

Auditoría

Todas las acciones realizadas por los usuarios quedan registradas en el sistema para poder realizar auditorías en el caso de ser necesario. Toda acción tiene relacionada la cuenta de usuario con la que la misma fue efectuada, además de la fecha y hora correspondientes.

Un caso especial de registro de auditoría es aquel en el que un usuario realiza una modificación sobre una entidad del sistema. En este caso, aquellos campos que sean modificados serán registrados en formato JSON, guardando el valor que tenía antes de la modificación y el nuevo valor. Un ejemplo de la modificación de una persona, donde su nombre y apellido han sido cambiados, tiene la siguiente forma:

Campo	**Descripción**	**Valor**
description	Cadena de caracteres con un resumen de la acción realizada.	La persona Perez, Juan ha sido modificada
user_id	Identificador del usuario que realizó la acción.	1
subject_type	Cadena de caracteres con el tipo de entidad sobre la cual se realizó la acción. Si la acción no se realiza sobre una entidad en particular estará vacío.	App\Persona
subject_id	Identificador de la entidad sobre la cual se realizó la acción. Si la acción no se realiza sobre una entidad en particular estará vacío.	130

properties	JSON con los atributos modificados de la entidad sobre la cual se realizó la acción. Si la acción no se realiza sobre una entidad en particular estará vacío.	{
		"original": {
		"apellido": "Perez",
		"nombre": "Juan",
		},
		"nuevo": {
		"apellido":"Perez Gonzalez",
		"nombre": "Juan Matías",
		}
		}

created_at	Timestamp del momento en el que la acción fue realizada.	2020-07-02 19:15:02

Anexo II

Control de acceso portuario, una solución utilizando tecnologías IoT

Felipe Evans[1], Hernan Hinojal[2], Gabriel Prieto Cassano[3], Bruno Colautti[4], Carlos A. Rico[5]
Grupo de Investigación en Desarrollos Informáticos, Universidad Nacional de Mar del Plata, facultad de ingeniería
[1]fevans@fi.mdp.edu.ar; [2]hhinojal@fi.mdp.edu.ar, [3]gabrielprietocassano@gmail.com, [4]bcolautti55@gmail.com, [5]carlos@fi.mdp.edu.ar

Resumen

El presente trabajo expone un estudio de caso de la solución de software propuesta a la problemática de control de acceso a la zona portuaria de una importante ciudad de la provincia de Buenos Aires. El desarrollo tiene una arquitectura distribuida con elementos del patrón publish/subscribe y almacenamiento tradicional en base de datos relacional. También se diseñó hardware específico para controlar los dispositivos de ingreso de personas y vehículos. Las herramientas de desarrollo utilizadas fueron de licencia totalmente libre (GNU) y se encuentran en repositorios abiertos a la comunidad. La implementación del mismo se encuentra operativa en la actualidad.

Introducción

La gestión de la seguridad en zonas portuarias es un problema de difícil gestión a nivel mundial. Debe permitir que un conjunto importante de personas, vehículos y máquinas accedan diariamente a zonas de trabajo de forma ágil, pero controlando potenciales actividades delictivas o de contrabando.

Según la Comisión Económica para América Latina y el Caribe (CEPAL) "la protección portuaria es un componente esencial de la vulnerabilidad económica del sistema de transporte marítimo de la Región de las Américas, y de la competitividad internacional. La misma debe contribuir a los programas generales de lucha contra el crimen tendientes a combatir el terrorismo y otras amenazas, como el tráfico ilícito de drogas, armas y personas y otras formas de crimen organizado, así como otros ilícitos que afecten la seguridad de la carga y el tráfico marítimo (robos, polizones, contrabando, entre otros), constituyendo una amenaza de explotación ilegítima de los puertos" [1]. En los últimos años, si bien la actividad económica ha mermado, el tránsito de personas y mercaderías por los puertos de la provincia de Buenos Aires es intenso y amerita que los puertos tengan soluciones de software y hardware modernas y acordes a las tecnologías actuales.

En el presente trabajo en particular, quedó claro desde el principio que para satisfacer las necesidades del Consorcio Portuario (el cliente) se debía adoptar una solución distribuida ya que la zona a controlar cuenta con distintos accesos dispersados en una amplia zona geográfica. Adicionalmente, había que permitir el paso de personas y distintos tipos de vehículos utilizando tarjetas de proximidad con una tecnología ya existente.

Como requisito, el consorcio solicitó que el sistema y los datos sean propios, y tener independencia de proveedores de electrónica. Otro requisito fue que sea de fácil instalación y mantenimiento. El gran problema de las soluciones comerciales actuales de acceso es que las mismas están atadas al hardware del proveedor, haciendo que cuando se quiere cambiar de fabricante sea necesario cambiar de software obligando a una migración indeseada, perdiendo en el mejor de los casos solo tiempo.

Este proyecto se enmarca en un convenio de colaboración entre el consorcio portuario regional Mar del Plata y la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata

Marco teórico

Para el desarrollo de la solución, fue necesario utilizar elementos de sistemas distribuidos. En palabras de Coulouris et al. un sistema distribuido es aquel en el cuál tenemos componentes comunicados por una red de computadoras coordinando sus acciones mediante el intercambio de mensajes [2]. Esto implica que a la vista del usuario, debe comportarse como un único sistema coherente, es decir obtener una vista de "sistema único". Esto realmente es algo difícil de lograr por lo que la

bibliografía opta por decir que solo “aparenta” ser un sistema único y coherente. Un sistema es coherente si se comporta de acuerdo a las expectativas del usuario. Ampliamente se denomina como un sistema único y coherente si el conjunto de nodos actúa de la misma manera, sin importar dónde, cuándo y cómo se produce la interacción entre este y el usuario.

Si bien estaba claro que la solución debía ser distribuida, era necesario planear detenidamente la arquitectura del software a diseñar. La ieee define como arquitectura a la organización fundamental de un sistema encarnada en sus componentes, las relaciones de los componentes con cada uno de los otros y con el entorno, y los principios que orientan su diseño y evolución [3]. Las arquitecturas consideran la organización lógica de los sistemas distribuidos en componentes de software, y también son conocidas como arquitectura lógicas. Es común aceptar que el diseño o la adopción de una arquitectura resulta crucial para el desarrollo exitoso de sistemas grandes [4].

La restricción impuesta por el cliente de seguir operando aun sin red local, implicó la necesidad de que el acceso sea controlado por dispositivos que tuvieran capacidad de procesamiento y almacenamiento local por sí mismos. La solución utilizada fue construida con la tecnología Arduino. Arduino es una plataforma de electrónica de código abierto basada en hardware y software fácil de usar, en especial para los desarrolladores que no son de formación electrónica.

El almacenamiento primario de la aplicación se pensó en una base de datos relacional SQL, ya que gran parte de la operatoria del software es de gestión tradicional.

Existía cierto desacople referencial entre persistencia primaria (base de datos relacional) y los microcontroladores Arduino que gestionaban el acceso, lo que llevó a la decisión de diseño de utilizar un middleware orientado a mensajes. En particular un acercamiento tipo IoT con un broker MQTT como capa intermedia.

MQTT es un protocolo de conectividad máquina a máquina (M2M), o IoT . Se diseñó como un transporte de mensajes de publicación / suscripción sumamente ligero. Es útil para conexiones con ubicaciones remotas donde se requiere una pequeña huella de código y / o el ancho de banda de la red es muy importante [6]. La comunicación en MQTT consiste en clientes (publicadores) que se comunican con un broker (“intermediario").

Especificaciones funcionales a resolver

- Determinar si una persona está habilita a ingresar a la zona portuaria en un momento dado.
- Establecer si un vehículo se encuentra habilitado para ingresar.
- Dejar registro de todo evento importante.
- Generar alertas sobre irregularidades o vencimientos.

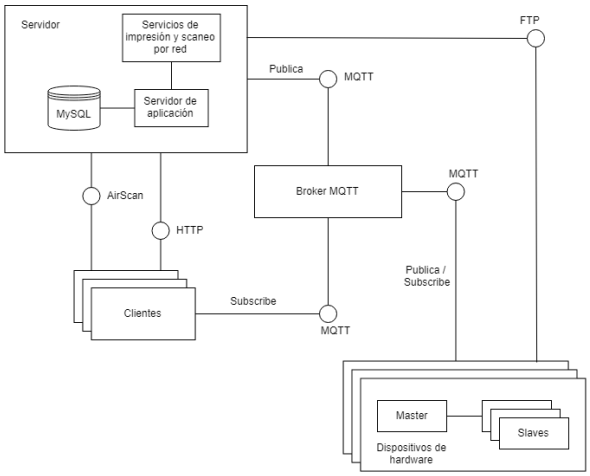
En cuanto a las propiedades del software, se pueden destacar:

- Escalabilidad horizontal al momento de agregar nuevas zonas restringidas.
- Permitir distintos tipos de sensores de acceso.
- Tolerancia a fallos, garantizando el funcionamiento del sistema incluso ante la indisponibilidad de la red local.
- La posibilidad de cambiar de proveedor de hardware.
- Una interfaz de usuario intuitiva y amigable.
- Un sistema extremadamente auditable.
- Rápida instalación.

Solución Propuesta

Para cumplir con el requisito de que los nodos puedan operar eventualmente de forma independiente y desacoplada, el control en el servidor de aplicación se implementó vía dos sistemas de comunicación independientes.

Por un lado la operación online, es decir con la red



funcionado, con intermediario tipo broker sobre protocolo de Internet (IP) con mensajes mediante el protocolo MQTT. Para el funcionamiento “desconectado”, se realizan transferencias por medio del protocolo de transferencia de archivos FTP. Para ello se definió una API de comunicación/tópicos.

Todo lo que tiene que ver con mensajes de control lo maneja el protocolo MQTT y aquello relacionado con descarga de las bases de datos de acceso o uplink de

bitácoras de auditoría de acceso de los equipos, por medio de archivos con protocolo FTP.

La API de Mensajes de control establecen los mensajes de: INICIO DE DISPOSITIVO, APERTURA DE PUERTA, CAMBIO DE CONTROL DE ACCESO, PING, PONG, SOLICITUD EXTRAORDINARIA DE RECARGA DE REGISTRO, SOLICITUD EXTRAORDINARIA DE ENVÍO DE BITÁCORA, SOLICITUD DEL SERVER A SYNC EL TIEMPO, ALARMA SILENCIOSA,CAMBIO DE REGISTRO DE UNA TARJETA. La estructura de datos elegida para estos mensajes fue JSON.

Como se definieron los mensajes de control también se estableció para traspaso de grandes volúmenes de información tablas de intercambio mediante el protocolo FTP. Para el sistema en el documento de definición de API se lo denomino MENSAJES DE INTERCAMBIO DE DATOS (FTP). Y se establecieron en principio tres tablas: TABLA DE TARJETAS HABILITADAS, ENVÍO DE REGISTRO DE ACCESOS. ENVÍO DE REGISTRO DE ACCIONES DE BOTONES. La estructura de datos de intercambio fue CSV.

Se establecieron dos estructuras de comunicación para los mensajes de control. Una donde los nombres de los campos son autodescriptivos y otro reducido. Por cuestiones de RAM de los dispositivos que se utilizaron la segunda opción o como se definió en el proyecto "Estructura reducida". Como se mencionó anteriormente la estructura elegida fue JSON.

Ejemplo del mensaje APERTURA DE PUERTA

```
{
    “N”: “XXXX”,
    “A”: “XXXX”,
    “C”:”INICIO”
    “F”: “XXXX”
}
```

Descripción de Campos

- N == Nombre: Nombre arbitrario del dispositivo.
- A == Area: Nombre del área que controla.
- F == Fecha: Fecha del dispositivo al momento del envío del mensaje. Este campo tiene como objetivo registrar el momento de inicio.
- C == Comando

Con respecto a todas las interfaces de clientes se eligió un esquema de "aplicación rica de internet" (ARI o RIA). Para esto se uso el framework de javascript Vue.js con un esquema de son sigle page, con comunicación con el servidor mediante AJAX.

Aunque se desarrolló el css de estilo para que se acomode al cambio de pantallas y sus resoluciones, al momento de armar el esquema de diseño y su testeo solo se desarrollo para pantallas de 10" o superiores con una resolución mínima de 1024x768.

Hardware

En cuanto al diseño del hardware se tuvieron tres grandes restricciones importantes : que cada microcontrolador ubicado en un punto de acceso al puerto pueda eventualmente operar como un nodo independiente y desconectado lo que implica que deben tener espacio suficiente para almacenar toda la base de datos de tarjetas a aceptar y bitácoras de acceso. Era necesario también que el hardware se pueda conseguir fácilmente desde Mar del Plata y pueda ser ensamblado e instalado por la mayor cantidad de empresas locales posibles.

Por estas restricciones se eligió usar tecnología de consumo de tipo Arduino, aunque se desarrollaron diseños propios de plaquetas para mejorar las fijaciones y agregar otros componentes más robustos a la hora del uso de algunos componentes como los relés.

Broker

Para funcionar como broker de mensajes se utilizó el software Eclipse Mosquitto, instalado en el servidor de aplicación. Mosquitto es de licencia open source implementando un broker de protocolo MQTT compatible con microcontroladores de bajo consumo hasta servidores de altas prestaciones.

Otro punto por el cual se eligió Mosquitto es porque soporta MQTT plano para la conexión del hardware de control y MQTT sobre WebSockets con TLS para que se conecten los cliente web.

Módulo de administración

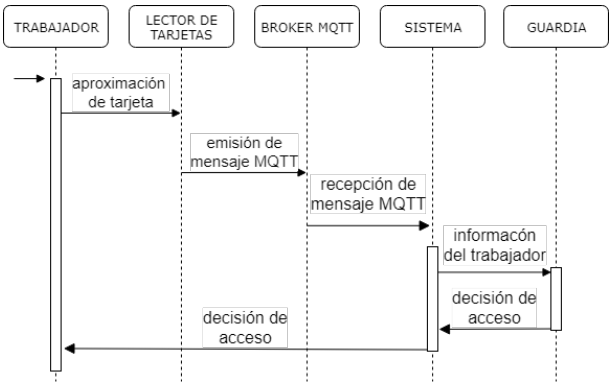
En este módulo permite la carga y la creación de personas, empresas y autos; como sus relaciones. Asimismo permite el control, almacenamiento y el seguimiento de toda documentación requerida para el acceso al puerto.

También en la interfaz tiene control de cámaras web para sacarle fotos a la persona, control de las impresoras de tarjetas como las impresoras de red comunes. Como así también los scanner de red.

Este módulo también está preparado desde el lado del backend para enviar mail de notificación a las empresas de toda documentación que vence o algún otro tema que se considera importante.

Módulo de guardia

El módulo de guardia es una interfaz web que permite al guardia visualizar toda tarjeta que se va pasando por los sensores de RFID asignados al sector que tiene que controlar. Estas visualizaciones se realizan a medida que van llegando los eventos de acceso recibidos mediante MQTT sobre WebSockets con TLS que genera una consulta AJAX al servidor para descargar los datos necesarios para el correcto control de la persona.



Existen dos interfaces: una pasiva y otra activa. La pasiva se diseñó para un televisor de 49" donde un guardia observa los datos de las personas que van ingresando o saliendo. La otra interfaz, "la activa" permite aceptar o negar el acceso, como también asentar cualquier observación sobre la persona, auto o en general que considere importante. En esa misma interfaz en el acceso de autos el guardia también mediante un lector "manual de RFID", que también está conectado por red, cargar los acompañantes y en forma simple cargar el auto con el que ingresa.

Este mismo módulo está preparado para que en un futuro reemplazar el RFID por el DNI. Para esto permite la lectura de todos los formatos del QR (PDF417) del DNI Argentino.

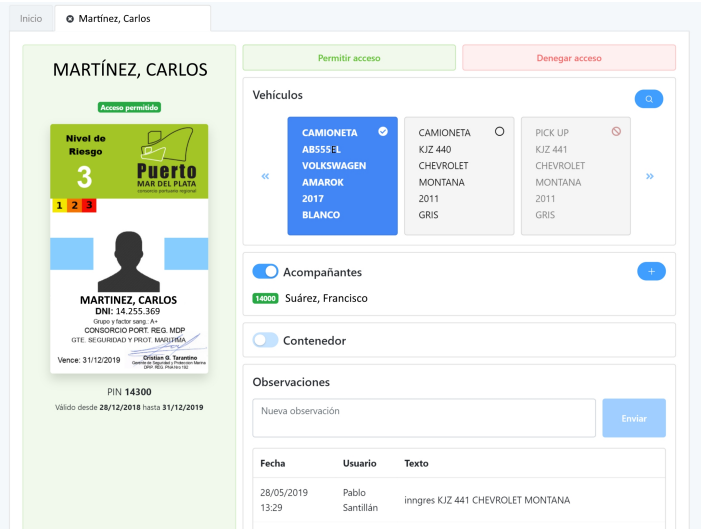
Módulo de OPIP

Este módulo es el de super usuario y el modulo de estadisticas.

Módulo de monitoreo

El módulo de monitoreo permite la visualización de todos los accesos de todas las puertas. A diferencia de guardias se muestra los accesos con distintos colores dependiendo el estado El estado más importante a controlar es cuando el sistema indica que la persona no tiene que entrar y el guardia le da acceso.

El guardia no ve las razones por las que el sistema no le da acceso. En este lugar pueden consultar la razón del no acceso. Por lo que, ante



un conflicto, el guardia puede llamar a esta central para que le indique los pasos a seguir.

Servicios Extras

El backend está escrito en PHP con una base de datos MySQL 5.7. La versión de base de datos MySQL 5.7 es necesaria ya que se requiere hacer uso de los campos JSON con índices sobre los mismos.

La emisión de material impreso se realiza con impresoras de red, tanto para imprimir tarjetas como documentación. El protocolo de comunicación que utiliza backend con las

impresoras es CUPS y AJAX entre el backend y la Cliente Web, pero esto es transparente al usuario. Desde la interfaz el usuario puede elegir la impresora que quiere y mandar a imprimir desde la interfaz web. Incluso pueden mandar a imprimir desde el otro edificio.

Con respecto a la documentación física, el sistema tiene un módulo que permite la digitalización de la misma. Por lo que se visualiza y controla lo que se escanea desde la interfaz web. También como en el caso de la impresión desde la interfaz se puede elegir el escáner. Para esto, se utiliza un esquema de control del escáner desde el backend mediante AirScan y entre el backend y la web mediante AJAX.

El sistema también le permite ver al guardia alguna cámara que considere crítica sobre la interfaz web. Para lograr esto se realiza mediante el protocolo HTTP Live Streaming (HLS) y la librería video.js. Hubo que resolver el problema que las cámaras solo permiten RTMP (***Real Time Streaming Protocol***) con lo cual en el servidor se programó un daemon con librerías FFmpeg para generar un servicio de conversión de HLS a RTMP.

Consideraciones de licencia e innovación

En lo referido a la innovación comercial, se desarrolló un producto propio de la empresa, a diferencia de las principales soluciones disponibles en el mercado que son ofrecidas mediante un contrato de licencia tipo “alquiler”. Además, la posibilidad de cambiar el fabricante de hardware les permite poder elegir al que, en tiempos o precios, mejor los satisfaga.

En cuanto a la innovación tecnológica de sistemas de similar funcionalidad o comerciales que se evaluaron, de distingue la forma de comunicación entre sistemas que en nuestro caso permite comunicar dispositivos de hardware con un cliente web de manera directa, sin la necesidad de instalar controladores para cada nuevo dispositivo que se desea agregar. Para lograr esto se utilizaran tecnologías emergentes usadas en IOT como MQTT. O por ejemplo el uso del protocolo AirScan logra que un programa pueda acceder comunicarse directamente a un scanner de red sin utilización de drivers, lo que posibilitó el desarrollo de una solución distribuida que dejando al usuario controlar cualquier escáner de red desde un navegador web. Y el protocolo IPP para centralizar la impresión de credenciales (controlando mejor lo que se imprime), y su correspondiente framework ui en javascript para web, logrando que que el usuario use cualquier impresora autorizada.

Con la finalidad de crear un sistema web rico y moderno, se utilizarán algunas de las tecnologías más extendidas en la actualidad. Estas tecnologías brindan mayor seguridad y validación sobre los datos ingresados al sistema, así como la integridad de los mismos, y además, dan la posibilidad de crear un cliente web con características a la par de las de una aplicación nativa del sistema operativo.

Como una innovación al proceso, se generarán reglas de auditoría para buscar anomalías de accesos. Para alcanzar dicho fin, se crearán bots.

Conclusiones

A nuestro juicio, el proyecto supone un aporte sustantivo para el control de acceso portuario, un área donde hay pocos proveedores y con soluciones propietarias. Incorpora innovación en el sentido de incluir elementos de IoT con interacción publish-subscribe. También ofrece una gran ventaja para el cliente ya que el código fuente del mismo está disponible en repositorios públicos y las herramientas de desarrollo y base de datos no requieren de pago de licencias. Es importante también destacar que se pudo realizar con las restricciones presupuestarias y de disponibilidad de recursos propios de trabajar entre una universidad con otro organismo de gestión pública. Más allá de los inconvenientes al momento de la implementación, por los problemas con diversos proveedores de electrónica (generados por las dificultades financieras en el país) y algunos problemas al momento de la compra por ser ambos entes públicos. La organización donde está en operación el software quedó satisfecha con el resultado en estos primeros meses de funcionamiento.

Trabajos Futuros

Si bien el sistema brinda información de auditoría e informes respecto a todo lo relacionado con el ingreso y egreso de personas y vehículos, el análisis con el ojo de experto se sigue realizando por seres humanos. Está en progreso el diseño de una funcionalidad que utiliza técnicas basadas en Bots para la búsqueda de patrones y anomalías en las bitácoras de acceso, para colaborar y asistir en la función de los guardias de acceso.

Aprovechando que el sistema maneja la digitalización de documentos se está pensando en en ir avanzando en despapelizar o eliminar directamente el soporte papel en la gestión de la organización. Por lo que en una segunda etapa se va a trabajar en buscar procedimientos que garanticen un igual o mejor funcionamiento de los circuitos administrativos, con procedimientos electrónicos.

Referencias

[[1] Sanchez J R., Garcia Bernal R, Manosalva Osorio, Rezende Sydney , Sgut M., PROTECCIÓN MARÍTIMA Y PORTUARIA EN AMÉRICA DEL SUR. Implementación de las medidas y estimación de gastos. IIRSA-CEPAL. 2004. Santiago de Chile. pp. 5.

[2] G.Coulouris, J. Dollimore y T. Kindberg, Distributed Systems, 5ta edición, 2012, Ed. Addison-Wesley ISBN 84-7829-049-4

[3] ISO/IEC/IEEE 42010

[4] Van Steen, M., Tanenbaum A. Distributed systems principles and paradigms. 3nd edition. 2017. Edición electrónica. http://www.distributed-systems.net

[5] https://www.arduino.cc/en/Guide/Introduction

[6] Banks A., Briggs E., Borgendale K., Gupta R., Estándar MQTTv.5, Marzo 2019, https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[7] Eclipse Mosquitto™. https://mosquitto.org/ . Visto el 5/SEP/2019

[8] Video.js. https://videojs.com/. Visto 5/SEP/2019.