UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE

"DISEÑO E IMPLEMENTACIÓN DE RECONOCIMIENTO DE PATENTES PARA PRODUCTO EASY ACCESS"

BENJAMIN AARON MIRANDA NARVAEZ

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Profesor Guía: Cecilia Reyes Covarrubias
Profesor Correferente: Luis Hevia Rodríguez

Septiembre - 2021

DEDICATORIA

Para mí y para ti.

AGRADECIMIENTOS

A mis amigos, familia, profesores, compañeros de trabajo y todas las personas que he conocido en el camino que de algún modo me han enseñado o me han hecho mejor persona. A todas las personas que me insistieron que terminara mi memoria. A las personas que no querían tocar el tema, pero me miraban feo. A las oportunidades que me han dado laboralmente sin tener un título.

RESUMEN

En esta memoria se abordó la integración de una API (Application Programming Interface) para el reconocimiento de patentes en la aplicación Easy Access de control de acceso de personas a edificios. Easy Access nace en el marco de la Feria de Software 2018 de la Universidad Técnica Federico Santa María, y con la presente memoria se amplían sus potencialidades, agregando un nuevo módulo para la gestión de accesos de vehículos.
Se hizo una búsqueda de un API que tuviera las mejores prestaciones, evaluando la cantidad de países para los cuales está disponible y los tiempos de respuesta. Se elaboró una aplicación móvil utilizando Flutter desde la que el usuario puede activar o desactivar accesos a patentes.
Palabras clave: Reconocimiento de pantentes – API – Flutter – Aplicación móvil.

ABSTRACT

This work is about an API integration for license plate recognition. As a new module for access control on Easy Access app. A search has been made to find the best api, taking in consideration from one part the number countries available and response time. Also, a mobile app was developed using Flutter framework, where users can activate or deactivate access privileges from their license plate stored.

Keywords—License Plata Recognition – API – Flutter – APP

GLOSARIO

API: Application Programming Interface.

Back End: Capa invisible para el usuario que posee las lógicas de negocio y conexión con bases de datos.

Dashboard: Tablero que muestra un resumen de una aplicación con información relevante al sistema.

Endpoint: Interfaz de comunicación expuesta por el backend.

Feedback: Retroalimentación.

Flutter: Es un SDK de código fuente abierto de desarrollo de aplicaciones móviles creado por Google. Suele usarse para desarrollar interfaces de usuario para aplicaciones en Android, iOS y Web.

Framework: Es un conjunto estandarizado de conceptos, prácticas y criterios.

Front End: Capa visible de las aplicaciones móviles o web, correspondiente a la interfaz de usuario.

Hot reloading: La capacidad de un sistema para actualizarse cuando detecta cambios.

INE: Instituto Nacional de Estadística.

IPSA: Índice de Precios Selectivo de Acciones

Login: Acción de iniciar sesión en un sistema.

Onboarding: Proceso de incorporación a un sistema, se suelen implementar para informar a los usuarios nuevos sobre el uso de la aplicación y llevarlos a la acción de registrarse.

QR: Quick Response code.

REST: Patrón de arquitectura web, Representational state transfer.

SDK: Software Development Kit.

UML:Unified Modeling Languages.

UTFSM: Universidad Técnica Federico Santa María.

WebView: Es un sistema que permite el embebido de un sitio web en una aplicación de Android.

PoE: Power over Ethernet.

INDICE DE CONTENIDOS

RESUMEN 4

ABSTRACT 4

INDICE DE FIGURAS 8

INDICE DE TABLAS 9

INTRODUCCIÓN 10

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA 11

1.1 IDENTIFICACIÓN DEL PROBLEMA 11

1.2 OBJETIVOS DE LA MEMORIA 14

1.2.1 OBJETIVOS PRINCIPAL 14

1.2.2 OBJETIVOS ESPECÍFICOS 14

1.2.3 ALCANCE DE LA MEMORIA 14

CAPÍTULO 2: ESTADO DEL ARTE 15

2.1 SISTEMAS DE CONTROL DE ACCESO VEHICULAR 15

2.2 SISTEMAS DE DETECCIÓN DE PATENTES 15

2.2.1 TÉCNICAS DE PREPROCESAMIENTO 16

2.2.2 DETECCION DE PLACA PATENTE 18

2.2.3 SISTEMAS DE DETECCION 19

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN 21

3.1 METODOLOGÍA DE DESARROLLO 21

3.2 TIPOS DE USUARIOS 21

3.3 HÍSTORIAS DE USUARIOS 23

3.4 HERRAMIENTAS Y FRAMEWORKS DE DESARROLLO 25

3.5 ARQUITECTURA DE LA SOLUCIÓN 25

3.6 MODELO DE DATOS 28

3.7 DISEÑO DE INTERFAZ DE USUARIO 29

3.7.3 VISTA HOME 30

3.7.5 VISTA AGREGAR NUEVA PATENTE 32

CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN 34

4.1 IMPLEMENTACIÓN CAPA BACKEND 34

4.2 IMPLEMENTACIÓN API DE RECONOCIMIENTO DE PATENTE 34

4.3 IMPLEMENTACIÓN DE INTERFAZ DE USUARIO 35

4.4 IMPLEMENTACIÓN DASHBOARD ADMINISTRADOR 35

4.5 DESARROLLO DE PROTOTIPO MÓDULO FÍSICO 35

4.6 INTEGRACIÓN DE MÓDULO FÍSICO AL SISTEMA 36

CAPÍTULO 5: VALIDACIÓN DE LA SOLUCIÓN 37

5.2 PRUEBAS PERFORMANCE API BACKEND 37

5.3 PRUEBAS FUNCIONALES APLICACIÓN MÓVIL 39

CAPÍTULO 6: CONCLUSIONES 49

REFERENCIAS BIBLIOGRÁFICAS 53

INDICE DE FIGURAS

Figura 1: Gráfico IPSA Histórico entre 2010 a 2020. Fuente: Elaboración propia, datos extraídos desde S&P Dow Jones Índices LLC 12

Figura 2: Primer paso LBP constructo de textura.Fuente (Rosebrock, Local Binary Patterns with Python & OpenCV, 2015) 16

Figura 3: Cálculo LBP píxel central. Fuente (Rosebrock, Local Binary Patterns with Python & OpenCV, 2015) 17

Figura 4: El valor del LBP calculado se guarda en una matriz.Fuente (Rosebrock, Local Binary Patterns with Python & OpenCV, 2015) 17

Figura 5: Diagrama funcionamiento operador patrones binarios locales. Fuente: Study of Local Binary Patterns, Tobias Lindahl 17

Figura 6: Diagrama detección placa patente. Fuente: Elaboración propia. 18

Figura 7: Diagrama de contexto del sistema. Fuente: Elaboración propia 26

Figura 8: Diagrama contenedor sistema Easy Access. Fuente: Elaboración propia 27

Figura 9: Diagrama de arquitectura. Fuente: Elaboración propia. 27

Figura 10: Modelo de datos. Fuente: Elaboración propia. 28

Figura 11: Diseño Vista Onboarding. Fuente: Elaboración propia 29

Figura 12: Diseño Vista Login. Fuente: Elaboración propia 30

Figura 13: Diseño Vista Home. Fuente: Elaboración propia. 31

Figura 14: Diseño Vista Patentes. Fuente: Elaboración propia. 32

Figura 15: Diseño Stepper parte 1, Flujo nueva patente. Fuente: Elaboración propia 33

Figura 16: Diseño Stepper parte 2, Flujo nueva patente. Fuente: Elaboración propia 33

Figura 17 : Esquemático y Render de la manufactura del empaquetado.Elaboración propia. 36

Figura 18: Gráfico de resultados, tiempo de respuesta vs cantidad solicitudes de usuario. Fuente: loader.io. 38

Figura 19: Gráfico de llamadas exitosas, erradas o time out vs tiempo. Fuente: loader.io 39

INDICE DE TABLAS

Tabla 1: Inversión en construcción categorizada por sector. Fuente: Cámara Chilena de la Construcción..... 13

Tabla 2: Tipos de usuarios. Fuente: Elaboración propia 21

Tabla 3: Componentes electrónicos módulo físico. Elaboración Propia. 36

Tabla 4: Configuración prueba de carga. Fuente: Elaboración propia. 38

Tabla 5: Plan de prueba aplicación móvil. Fuente: Elaboración propia 39

INTRODUCCIÓN

Boldware es un emprendimiento de 4 sansanos, formado en el contexto de la Feria de Software 2018 de la Universidad Técnica Federico Santa María. Con su primer producto Easy Access, busca transformar los accesos de edificios a la era digital.

Easy Access actualmente permite que residentes de un condominio puedan gestionar listados de invitados y con la ayuda de reconocimiento facial estos ingresen de manera simple y rápida a edificios. Incorpora al software un módulo físico de reconocimiento facial que permite su integración con sistemas de acceso de diversa índole.

Su ecosistema considera además del módulo, una aplicación para residentes, otra para conserjes y, finalmente, un ***dashboard*** para administradores de edificios. Estas interactúan entre sí mediante notificaciones.

Easy Access no solo viene a resolver problemas de acceso si no que también de trazabilidad del proceso de ingreso a edificios. Hoy en día la seguridad recae básicamente solo en un conserje que muchas veces, dada la monotonía de su labor, no lleva controles tan exhaustivos.

Por otro lado, dentro de los métodos actuales de control de acceso a estacionamientos se utilizan tarjetas, controles remotos o el tradicional método manual. Bajarse del auto implica un gran riesgo. Sin entrar mucho en detalles según un reportaje de prensa, más de 700 robos de vehículos ocurrieron durante los primeros 5 meses del año 2018, reflejando un alza de 50% respecto al año 2016 que fue donde comenzó a llevarse registro de dicha estadística.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 IDENTIFICACIÓN DEL PROBLEMA

Durante el 2019 según el boletín "Sectores Económicos, permiso de edificación autorizada" elaborado por el INE[1], la cantidad de m2 autorizados para la construcción de edificios destinados a vivienda se incrementó en un 29,1% (INE, 2019). Los motivos de este aumento en la demanda de lugares habitacionales pueden ser diversos, desde la llegada de inmigrantes al país, el déficit de viviendas para chilenos, el interés de inversionistas de contar con departamentos para ofrecer hospedajes con la llegada de aplicaciones como Airbnb.

Esto trae como consecuencia directa un aumento en el tránsito de personas en los condominios, generando la necesidad de robustecer los sistemas de control de acceso tanto de personas como de vehículos a lugares residenciales, buscando siempre la comodidad.

Por otra parte, en ciudades como Santiago, los estacionamientos de los edificios son arrendados por minuto, horas, meses o transados por hasta 900 UF la unidad, siendo un gran negocio para los residentes.

Controlar el flujo de quienes hacen ingreso se vuelve entonces una tarea que impacta directamente en el uso de recursos, tanto humanos, tecnológicos como monetarios convirtiéndose en un tema relevante a la hora de administrar cualquier tipo de edificio. Además, esta necesidad de control tiene un alcance global, por lo que contar con un servicio tecnológico de control, no tiene límites para escalar a un uso internacional.

Pero no basta con el control, también tiene que ser relativamente sencillo de adoptar y fiable.

Empresas como Groupon Latam, manejan cierto stock de estacionamientos para sus trabajadores que son entregados por el edificio corporativo en que arriendan oficinas, manteniendo una plaza fija para sus trabajadores. Estos deben diariamente marcar una tarjeta en el estacionamiento y llevar personalmente el comprobante para su posterior cobro a la secretaría de la empresa.

También debemos tener en consideración aspectos de higiene, sobre todo en el escenario actual con la pandemia del coronavirus, que para julio del año 2021 ha afectado a cerca de 20 millones de personas y cobrado la vida de más de 4 millones de personas.

Gran parte de las medidas para el control de la pandemia incluyen el distanciamiento social, así lo han demostrado países que han sido particularmente efectivos en mantener bajas tasas de contagio, como por ejemplo Nueva Zelanda 2767 casos detectados, o Japón con

[1] El INE es el acrónimo de Instituto Nacional de Estadísticas.

menos del 1% de su población contagiada[2]. Si bien este no es el único factor de éxito sigue siendo uno de los más relevantes.

Es así que un sistema de control que reduzca la interacción entre personas es crítico, dado que una de las formas más efectivas de controlar el crecimiento de la tasa de contagio es precisamente el distanciamiento social.
La pandemia también es económica. El gran confinamiento ha provocado caídas drásticas en las bolsas de diversos países frente a la creciente incertidumbre. Sin ir más lejos en Chile el IPSA[3], se contrajo hasta los $2045 en abril de 2020, esto implica un gran retroceso en materia económica. Como se puede dilucidar en la gráfica de la Figura 1, el índice ha repuntado en los meses posteriores, pero a valores considerablemente inferiores que al comienzo de la década.

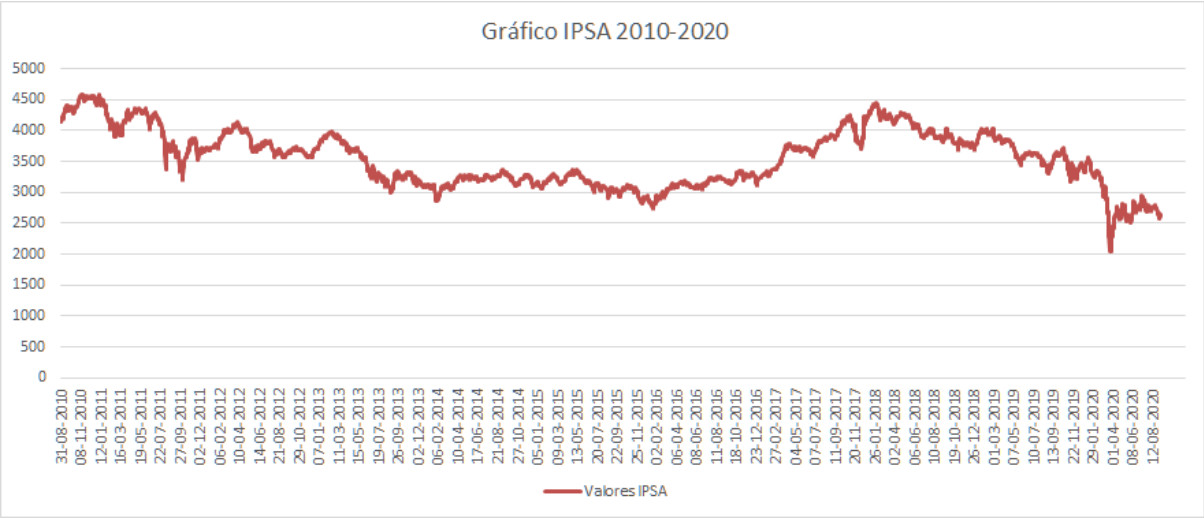

Figura 1: Gráfico IPSA Histórico entre 2010 a 2020. Fuente: Elaboración propia, datos extraídos desde S&P Dow Jones Índices LLC

Durante el 2020, se han percibido también efectos adversos a nivel de inversión en el sector de la construcción. Como se observa en la Tabla 1, esta se contrajo en el sector privado en un 20% mientras que en la pública aumentó en tan solo un 2% para la categoría de vivienda.
El panorama global tampoco luce muy alentador, si se compara la inversión de la categoría vivienda e infraestructura con la respectiva del año 2019, pues notamos una baja correspondiente a un 10%, caídas que no se registraban desde hace 37 años.

[2] Datos a Julio 2021
[3] El IPSA corresponde al Índice de Precio Selectivo de Acciones y es el principal índice bursátil de Chile.

INVERSIÓN EN CONSTRUCCIÓN			
SECTOR	2018	2019	2020
	Millones de UF		
VIVIENDA	228,9	236.8	205.5
Pública	**45,0**	**47,6**	**48,8**
Privada	**183,9**	**189,3**	**156,7**
Copago prog. Sociales	34,1	33,1	29,7
Inmobiliaria sin subsidio	149,7	156,2	127,0
INFRAESTRUCTURA	421,5	450,4	409,7
Pública	**165,7**	**171,4**	**195,7**
Pública	113,1	115,6	131,1
Empresas autónomas	38,1	39,9	44,1
Concesiones OO.PP.	14,6	15,8	20,5
Productiva	**255,8**	**279,0**	**214,0**
EE.Pública	19,4	15,7	10,7
Privadas	236,4	263,4	203,3
INVERSIÓN EN CONSTRUCCIÓN	650,4	687,2	615,2

Tabla 1: Inversión en construcción categorizada por sector. Fuente: Cámara Chilena de la Construcción

Esta caída en el sector se debe a la detención de las obras tanto en fase de ejecución como de iniciación. Específicamente en el sector público también se atribuye dicha baja a la reasignación en el presupuesto realizado para financiar el Plan de Emergencia sanitaria por parte del Gobierno.

Sin embargo, se espera que entre 2021 y 2022 la inversión se recupere y vuelva a su crecimiento normal de entre 4 y 5% anual, lo que mantiene el mercado atractivo.
Easy Access tiene el potencial de no solo mejorar la experiencia de usuario si no que también su seguridad e higiene. Brindando al usuario un confort y robustez, que hoy en día métodos alternativos no poseen. Los controles se quedan sin batería, las tarjetas se pierden, mientras que las placas patentes siempre deberían estar en su lugar.

1.2 OBJETIVOS DE LA MEMORIA

1.2.1 OBJETIVOS PRINCIPAL

Desarrollo de un sistema de detección para placas patentes chilenas integrado al software de control de accesos a edificios Easy Access. De esta manera además de utilizar reconocimiento facial para peatones, podrá vincularse al control de acceso a estacionamientos de edificios residenciales, mejorando así tanto la cobertura de su servicio como la experiencia de usuario de quienes hacen ingreso mediante vehículo y la seguridad de los residentes.

1.2.2 OBJETIVOS ESPECÍFICOS

- Diseño, desarrollo e integración de un API de reconocimiento y detección de placas patentes chilenas.
- Desarrollo de prototipo módulo físico de detección de placas patentes, integrado al sistema de información y notificación de eventos para control de acceso a estacionamientos de edificios.
- Diseño e implementación de sistema de información para el control y gestión de patentes en aplicación móvil de conserje y residente, con el fin de que puedan permitir o denegar el acceso a vehículos además de ser notificados de eventos ocurridos frente a módulo físico.

1.2.3 ALCANCE DE LA MEMORIA

La solución a implementar consistirá de un módulo de detección de patentes que interactúa con el sistema Easy Access, es decir los residentes que tengan derecho a uso de estacionamiento, podrán autorizar patentes para que puedan ingresar. Sin el uso de controles de por medio o interacción con otros medios.

El prototipo detectará la llegada de un vehículo hacia el portal del edificio y mediante visión computacional será capaz de detectar su patente; en caso de tener la autorización, el sistema le permitirá ingresar notificando al usuario de la aplicación o residente que el vehículo ha entrado en la propiedad.

Además, se le brindará al usuario la capacidad de ingresar uno o más vehículos, así como también restringir el ingreso con una lista negra. Asimismo, se podrán conceder permisos ocasionales, es decir fijar una fecha de expiración para este, llevar un catastro de qué patentes y cuántos vehículos transitan por el lugar, entre otra serie de estadísticas que pueden ser de utilidad.

CAPÍTULO 2: ESTADO DEL ARTE

2.1 SISTEMAS DE CONTROL DE ACCESO VEHICULAR

Actualmente existen una serie de sistemas de control de acceso para este propósito. Sin embargo, es necesario distinguir entre aquellos que son de uso público (se arriendan por horas o minutos) y aquellos de uso privado (en general, pertenecen a residentes de un edificio y se arriendan por mes).

En los del primer tipo, los principales sistemas existentes se centran en que no solo deben autorizar el ingreso si no que además incorporar un método de pago. Existen máquinas semi automáticas en que, a cambio de monedas o billetes, entregan un ticket que se debe entregar al momento de salir para levantar la barrera.

Estos sistemas son poco eficientes pues no permiten el procesamiento en paralelo de solicitudes, ni tampoco de forma remota, siendo el usuario del vehículo quien debe dirigirse hasta el sistema para generar el ticket. Otros utilizan un enfoque similar, solo que se debe interactuar al entrar, generando muchas veces atochamientos en el acceso de los estacionamientos.

En nuestro país, existe **Parkingapp**, que busca solucionar estos problemas mediante el uso de teléfonos móviles, códigos **QR** y métodos de pago dentro de la aplicación.

En países europeos existe **Nedap,** empresa que tiene el software **MOOV**, este sistema está pensado para ser utilizado por un único administrador que provee de permisos a vehículos de forma remota, se integra con dispositivos de detección de patentes que la misma empresa ha creado.

De parte de la detección, Nedap utiliza tarjetas UHF (Ultra High Frequency); cuenta con una antena que detecta la tarjeta ubicada a una distancia que varía entre 5 a 10 metros, según el modelo. Dicha tecnología forma parte de la plataforma **TRANSIT** de la compañía. Ellos también desarrollaron y comercializaron el ANPR Access V2 Short-Range que permite la captura y reconocimiento de patentes, según su ficha técnica se encuentra capacitado para reconocer patentes en todos los países del mundo. Aunque su eficacia está principalmente comprobada en Europa.

2.2 SISTEMAS DE DETECCIÓN DE PATENTES

En el mercado existe una variedad no menor de cámaras y dispositivos que permiten la detección de patentes, basadas en diversas técnicas. Algunos de estos serán listados y considerados por ser los que se asemejan más al prototipo a construir.

2.2.1 TÉCNICAS DE PREPROCESAMIENTO

El paso crítico a la hora de detectar patentes es el preprocesamiento de la imagen, es decir ubicar la zona de interés o la placa. Para esto se utilizan los siguientes procesos:

Gray scale conversion

Se convierte la imagen que ingresa al sistema de detección a escala de grises, es decir los colores rojo, verde y azul son separados de los 24 bit del código de color, para ser transformados a 8 bit usando la fórmula:

$$\boldsymbol{Imagen_gray} = 0.59\boldsymbol{R} + 0.30\boldsymbol{G} + 0.11\boldsymbol{B}$$

$$Donde\ \ \boldsymbol{R}\ (red), \boldsymbol{G}\ (green), \boldsymbol{B}\ (blue)$$

Median filtering

Este filtro evalúa cada píxel y sus 8 vecinos en una matriz de 3x3, utilizando la media de color gris de estos como nuevo valor del pixel, con esto se remueve parte del ruido de la imagen.

Se evalúa entonces cada sub matriz. Si el valor del centro comparado con los 8 vecinos es mayor que el número, entonces se cambia por un cero. Caso contrario si el número es menor, se reemplaza por un 1. Con esto se obtiene una imagen en escala de grises con menos información.

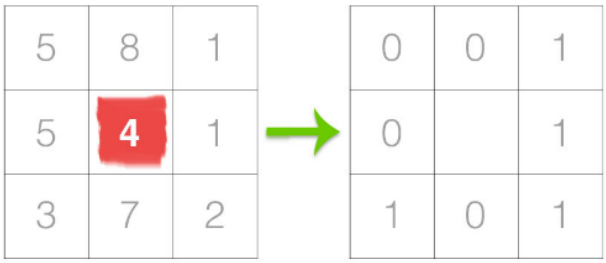

Figura 2: Primer paso LBP constructo de textura.Fuente **(Rosebrock, Local Binary Patterns with Python & OpenCV, 2015)**

Luego se calcula el valor del LBP del píxel central como se muestra en la Figura 3.

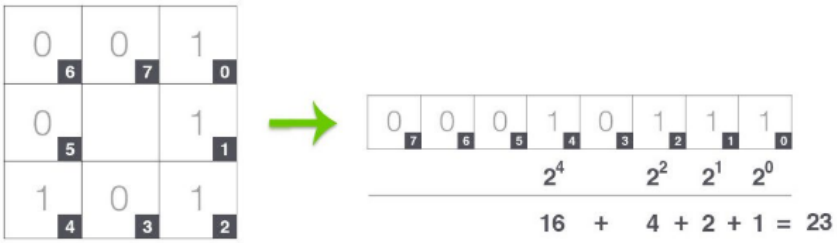

Figura 3: Cálculo LBP píxel central. Fuente **(Rosebrock, Local Binary Patterns with Python & OpenCV, 2015)**

Se almacena el valor del píxel central en la matriz (lado derecho) y se sigue iterando sobre cada pixel de la imagen de entrada (lado izquierdo)

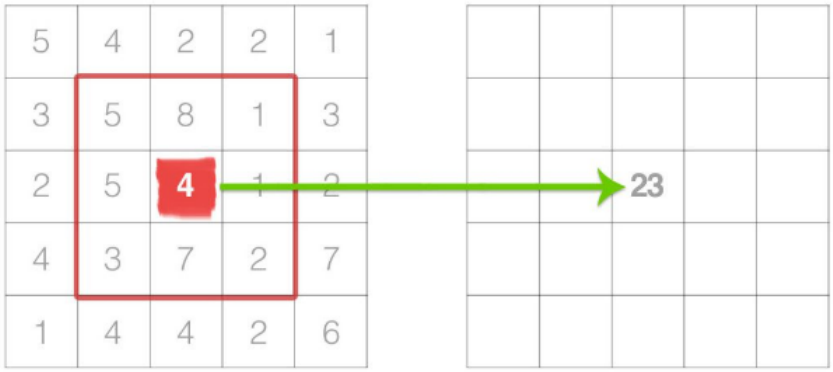

Figura 4: El valor del LBP calculado se guarda en una matriz.Fuente **(Rosebrock, Local Binary Patterns with Python & OpenCV, 2015)**

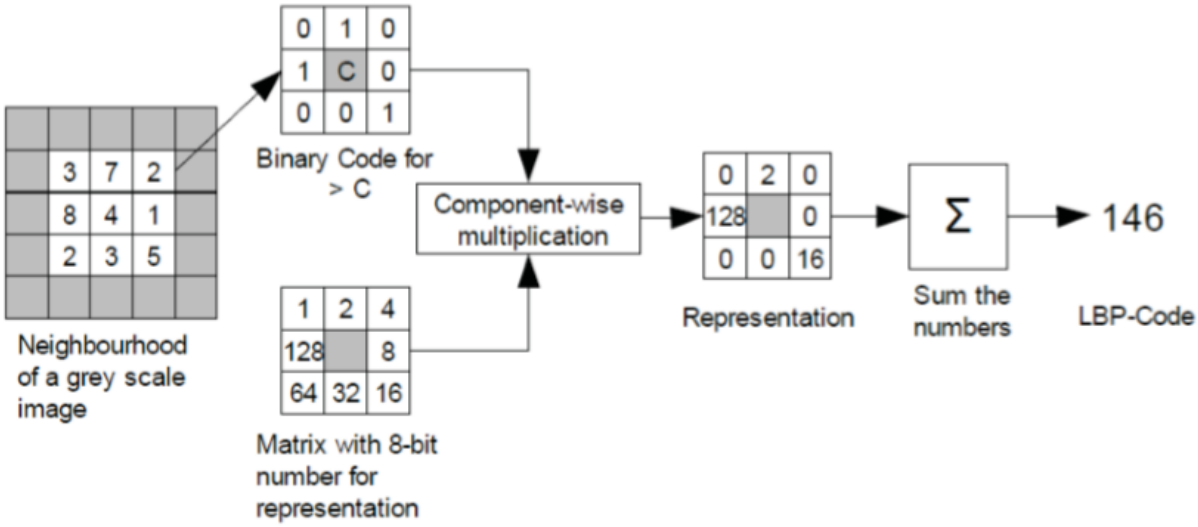

Figura 5: Diagrama funcionamiento operador patrones binarios locales. Fuente: Study of Local Binary Patterns, Tobias Lindahl

2.2.2 DETECCION DE PLACA PATENTE

En la sección anterior, se revisaron mecanismos que permiten mitigar el ruido de la imagen. Entendiendo como ruido todo aquello que no corresponde al área de interés, es decir la placa patente y sus caracteres.

Ya habiendo preprocesado la imagen, esta atraviesa el flujo que observamos en la Figura **6**.

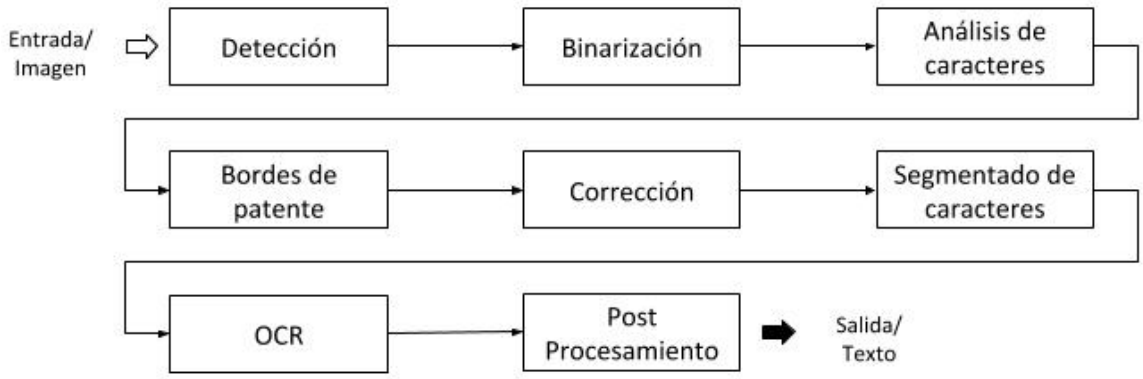

Figura 6: Diagrama detección placa patente. Fuente: Localization Techniques in ANPR Systems: A-State-of-Art, Pawan Kumar Dahiya, 2017.

¿Qué ocurre en cada fase?

Detección: Esta etapa filtra la imagen en búsqueda de la región de interés, es decir, la placa patente.

Binarización: Convierte la imagen a blanco y negro con el propósito de poder mitigar el ruido que agregan otros elementos de la imagen.

Análisis de caracteres: Busca todas las figuras que son aproximadamente similares a los tamaños de los caracteres de la patente. Comienza buscando los caracteres pequeños para luego explorar, gradualmente, los caracteres más grandes. De encontrar algo, lo almacena para continuar el análisis.

Bordes de la patente: En esta fase se encuentran los bordes de la patente. Se debe tener en cuenta que la fase de detección sólo es responsable de identificar una posible región donde puede existir una patente. A menudo va a proporcionar una región que es un poco más grande o pequeña que la dimensión real. En esta fase se trata de encontrar los bordes exactos de la patente ingresada.

Corrección: Dado los bordes de la patente, en esta etapa se trata de corregir la región de la patente a un tamaño estándar. Idealmente, esto dará una imagen de la patente correctamente orientada, sin rotación o inclinación.

Segmentación de caracteres: En esta fase de segmentación de caracteres, se trata de aislar todos los caracteres que componen la imagen de la patente. Utiliza un histograma vertical para encontrar los espacios que existen entre los caracteres de la patente. En esta fase también se limpia las cajas de personajes mediante la eliminación de puntos pequeños y desconectados de la región de los caracteres. También intenta eliminar las regiones del borde de la patente, para que no se clasifique de manera inapropiada como carácter.

OCR (Optical Character Recognition)**:** En esta fase, se analiza cada caracter independientemente. Para cada imagen de caracter, se calculan todos los caracteres posibles y sus niveles de confianza.

Post - Procesamiento: Dada una lista de todos los caracteres de OCR y coincidencias posibles, el post procesamiento determina las mejores combinaciones de letras de la patente. Está organizado como una lista N superior. El procesamiento posterior deshabilita todos los caracteres por debajo del umbral determinado.

2.2.3 SISTEMAS DE DETECCION

En el mercado ya existen algunas soluciones que abordan el problema. En general proveen el flujo completo, algunos incluyendo el hardware necesario para realizar la captura del vehículo y su patente. Estos por lo general son cerrados y vienen fuertemente acoplados con una alta dependencia entre el dispositivo y los algoritmos de detección.
A continuación, se listan algunos de los productos encontrados con mayor relevancia para este proyecto.

Openalpr

Es una biblioteca de código abierto escrita en C++ que permite a partir de una fotografía realizar todo el procesamiento hasta obtener la patente de un vehículo en texto. Tiene un *pipeline* que permite la corrección de posición de captura lo que es sumamente útil pues no siempre las capturas son alineadas al plano. Lo destacable es que tiene un 90% de confianza en la detección y que puede ser entrenado para el formato chileno, puesto que está en primera instancia diseñado para el europeo.

ANPR Access V2 Short-Range

Este dispositivo tiene la gran particularidad de adaptarse a sistemas de gestión de patentes mediante API Rest, cuenta con conexión a internet y una memoria de 4Gb de almacenamiento multipropósito. Está construido con una cámara de 2 megapíxeles y tiene un alcance de hasta 6 metros.

Cámara Lectura de Patentes DS-2CD4A26FWD-IZS/P Hikvision

La cámara tiene función día/noche con filtro IR CUT e incorpora LEDs infrarrojos con un alcance de hasta 50 metros, por ende lo convierte en un potente detector de patentes de autos durante todo el día. Está equipada con un grado de protección IP67 resistente a condiciones climáticas adversas, contra lluvia, viento y polvo.

POE (Power over Ethernet) integrado compatible con 802.3af reduce el problema de cableado, por lo que la solución de vigilancia es menos costosa en términos de implementación.

SecurOS AUTO

Es un sistema de gestión de video analítica de la compañía ISS [4](Intelligent Security Systems), que permite el reconocimiento y captura automático de patentes de vehículos a una velocidad límite de 210 Km/H.

Plate Recognizer

Es una API de reconocimiento de patentes que disponibiliza en su versión pagada un SDK, la ventaja de esta API es la capacidad global del servicio dado que está probada en más de 90 países. Inclusive tienen un caso de estudio diseñado para India, donde las patentes tienen una variedad no menor de formatos.

[4]ISS WEB - https://es.issivs.com/

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

3.1 METODOLOGÍA DE DESARROLLO

Se ha organizado el trabajo de desarrollo de las solución en 4 módulos: API reconocimiento de patentes, API backend, módulo físico y aplicación móvil.

Se trabaja en torno a historias de usuarios. Se realizan sprints con iteraciones de 2 semanas. Se realizó un primer sprint de una semana en que se montaron los entornos de desarrollo realizando algunas pruebas, la segunda semana culminó con una prueba de concepto probando el reconocimiento de patentes con capturas que se ingresan al sistema de forma manual. En siguientes iteraciones se incorporaran las lógicas de eventos de detección de vehículos para luego ser incorporados en el módulo físico y en la aplicación móvil.

3.2 TIPOS DE USUARIOS

Como todo proyecto de desarrollo de software, este cuenta con la presencia de distintos actores que interactúan con el dispositivo y la aplicación móvil construida. Fue de vital importancia identificar estos actores para conocer sus necesidades, de tal manera de asegurar el desarrollo de un software de calidad, toda vez que se ajuste a satisfacer exitosamente dichas necesidades.

En la Tabla 2 a continuación se detalla la descripción de cada rol o “tipo de usuario”.

Tabla 2: Tipos de usuarios. Fuente: Elaboración propia

Rol	**Descripción**
Residente	Habitante del inmueble, este puede ser arrendatario o propietario. Cuenta con acceso a la aplicación Easy Access, pudiendo autorizar o denegar acceso a usuarios invitados, notificar al conserje de potenciales visitas y agregar placas patentes tanto propias como de visitas.
Guardia/Conserje/Recepcionista	Persona que controla el acceso al edificio, cuenta con acceso a Easy Access. Recibe notificaciones de parte de residente. Puede registrar, identificar y revisar el log de uso del sistema.

Invitado	Persona que es agregada por el residente al sistema para que pueda hacer ingreso al recinto de forma automatizada. Puede hacer ingreso mediante la patente de su vehículo o vía reconocimiento facial.
Administrador	Persona a cargo de la gestión del edificio, cuenta con acceso a un dashboard donde se pueden observar métricas de uso y generar invitaciones para residentes, entre otras funcionalidades como denegar permisos de acceso.

3.3 HÍSTORIAS DE USUARIOS

Tabla 3: Tabla de historias de usuario. Fuente: Elaboración propia.

ID	Título / Descripción / Criterio de Aceptación
HU01	**Registrar patente en el sistema**
	Un usuario residente necesita ingresar mediante texto la placa patente del vehículo que desea permitir acceso.
	El usuario recibe la confirmación del procedimiento, notificando que su patente fue registrada correctamente en el sistema.
HU02	**Listar patentes pertenecientes al usuario**
	Tanto usuario administrador como usuario residente pueden visualizar el listado de patentes registradas.
	Dentro de la interfaz es posible revisar el listado de patentes y su estado actual (permitido o rechazado). Un residente no puede ver patentes de otro usuario, solo las que ha ingresado mientras que, el administrador puede listar todas las patentes existentes en el sistema.

HU03	**Editar patente**
	El usuario residente puede cambiar información asociada a las patentes que tenga en sistema.
	Al finalizar las modificaciones y presionar guardar, estas quedan en el sistema notificando al usuario del cambio

HU04	**Notificación patente reconocida**
	Cuando un invitado o usuario residente se acerca al tótem con su vehículo, este es detectado por el sistema.
	El usuario “dueño” de la patente es notificado del ingreso según el caso, además se abre la puerta del lugar para que pueda entrar al estacionamiento del edificio. Así mismo se recibe un feedback visual de parte del tótem.

3.4 HERRAMIENTAS Y FRAMEWORKS DE DESARROLLO

Montar un entorno de desarrollo que permita la ejecución del proyecto es un aspecto de vital importancia para el paso eventual desde desarrollo a producción. Es considerando esto que se utilizan contenedores de **Docker** para cada unidad del sistema, su versatilidad y la facilidad para migrar el contenido a otras plataformas, los hacen ideal para mitigar posibles problemas de compatibilidad con entornos como Amazon u otras plataformas con mayor capacidad de procesamiento que pueden llegar a ser requeridas cuando el uso de la aplicación escale.

Por otro lado, para el desarrollo de la interfaz de la aplicación se utiliza **Flutter**, un framework multiplataforma que permite el desarrollo de solo una aplicación y generar compilados para las dos plataformas principales (**iOS y Android**). Se exploraron otras opciones como Ionic o React Native o incluso Nativo. En el caso de Ionic 2 se descartó por presentar dificultades para integrar el uso de sensores de los dispositivos móviles. Se hicieron pruebas en React Native, pero el proceso de desarrollo en dicha plataforma resultó ser engorroso y con un sin número de inconvenientes. Cosas básicas como el **hot reloading** tienden a presentar bastante demora.

Mientras que **Nativo** implica duplicidad de esfuerzo al tener que elaborar una aplicación por cada plataforma. Un análisis de su rendimiento muestra que tiende a ser ligeramente inferior, tanto en fluidez, uso de recursos de CPU y memoria. Esto se debe a que **Ionic** utiliza **WebView**, este enfoque genera una aplicación con un sitio web embebido.

Para la capa de gestión de datos se utilizará **NodeJs**, principalmente por su soporte para la gestión de uso concurrente y **MongoDB** como base de datos.

Para elaborar el documento de diseño se utilizó historias de usuario y el modelo c4[5] en vez de **UML** para representar arquitectura de software. El modelo c4 proporciona cuatro niveles de abstracción, estos son Contexto, Contenedores, Componentes y Código. Se elaboró hasta el diagrama de componentes, pues el propósito es disponer de un mapa como guía para navegar el proyecto más que detallar las diferentes funciones de cada componente.

3.5 ARQUITECTURA DE LA SOLUCIÓN

Se utilizó un modelo mixto entre microservicios y monolítico. Por un lado, se tiene 2 microservicios, el de reconocimiento de patentes y el de rostro. Y por otro, la capa de datos que gestiona la información de usuarios, edificios y patentes, lo cual corresponde a un modelo monolítico.

Las aplicaciones móviles y web interactúan con la capa de backend mediante API REST. Esta última encapsula todos los comportamientos de las lógicas de negocio y de usuarios. A su

[5] El modelo c4 nace como respuesta a UML que tiende a ser engorroso y de difícil adopción en equipos modernos de trabajo. Más detalles en https://c4model.com/

vez esta capa interactúa con dos servicios externos, uno el servicio de MongoDB Cloud de base de datos y el segundo, el Plate Recognizer, además de la capa de reconocimiento facial.

Lo anterior es posible gracias al Proxy reverso que se encarga de enrutar las diferentes API en el servidor Cloud Scaleway. En la Figura 7 se puede observar la estructura que posee el servicio.

Esta se determinó en función de posibles integraciones que se lleven a cabo o el eventual reemplazo de los servicios de reconocimiento facial o de patentes. Esta arquitectura brinda la flexibilidad suficiente como para incluir nuevas tecnologías que puedan resultar más efectivas o costo-eficientes, pues no es necesario cambiar la manera en que interactúan las capas de frontend con backend, simplemente basta hacer los cambios desde el backend con el servicio que que se quiera integrar. Haciendo más fácil el mantenimiento de documentación del producto.

Se utilizó la tecnología de contenedores por su flexibilidad también a la hora de ejecutar, escalar o migrar servicios de plataforma.

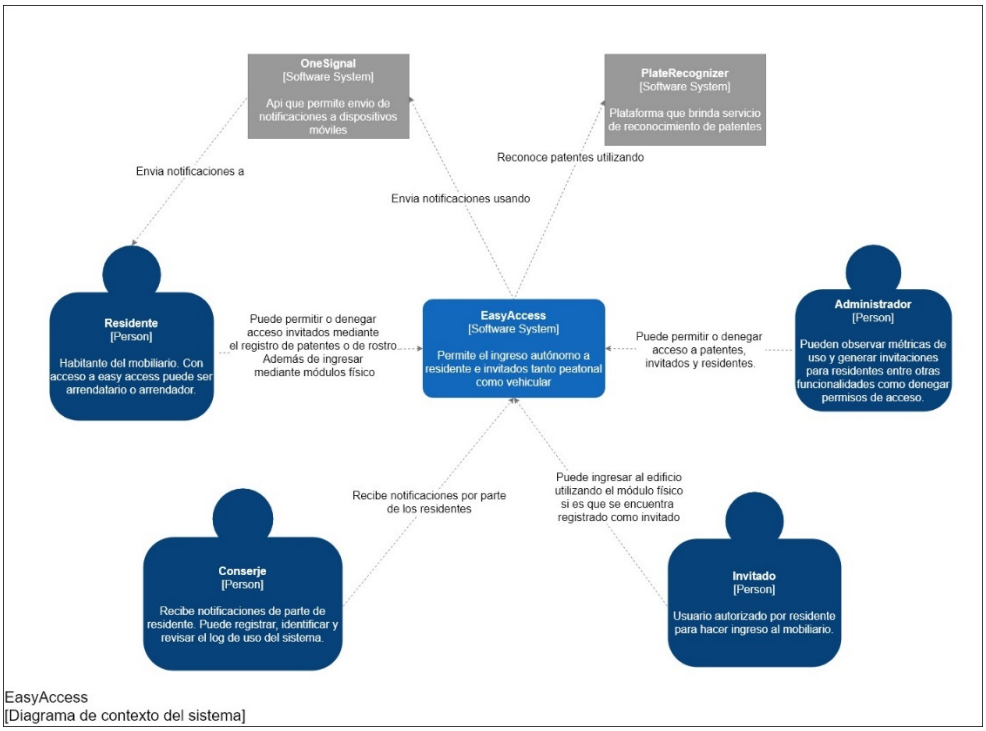

Figura 7: Diagrama de contexto del sistema. Fuente: Elaboración propia

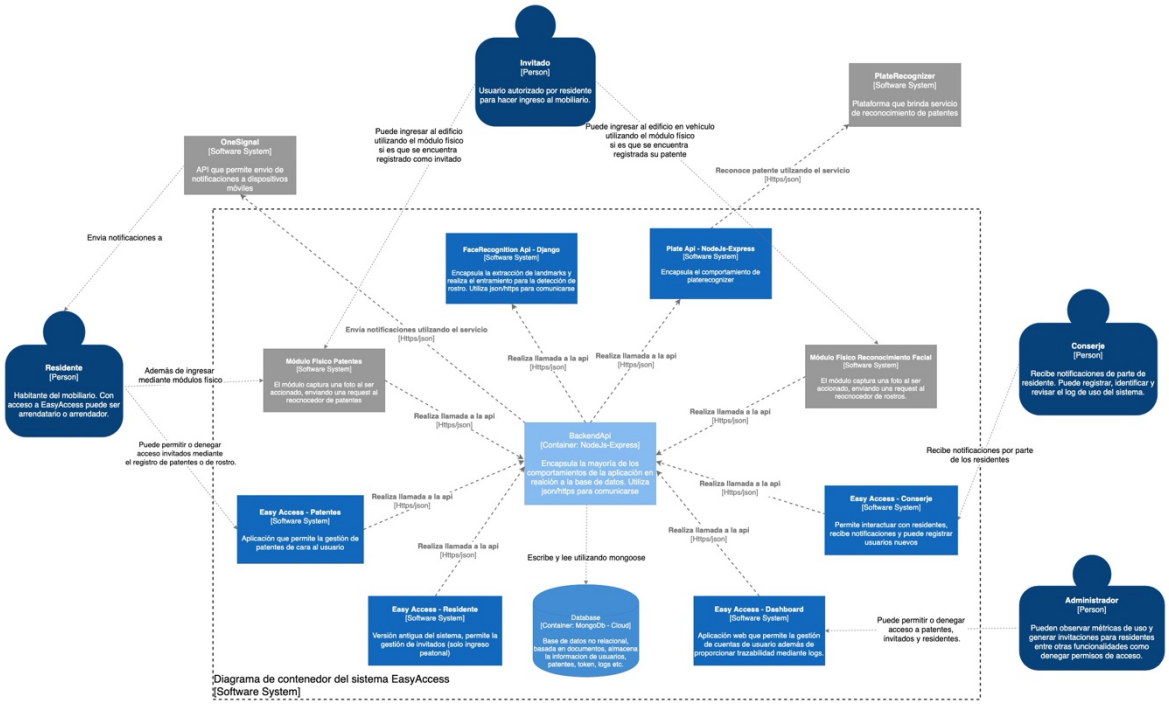

Figura 8: Diagrama contenedor sistema Easy Access. Fuente: Elaboración propia[6]

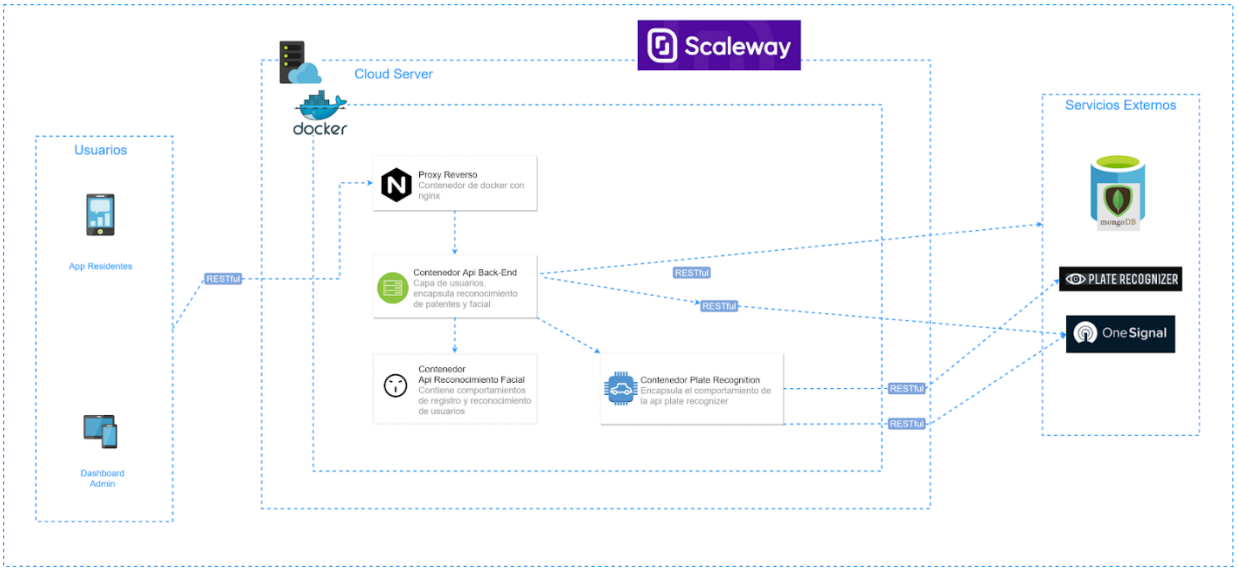

Figura 9: Diagrama de Arquitectura. Fuente: Elaboración propia.[7]

[6] Puedes visualizar en el siguiente link una versión con más detalles: https://drive.google.com/file/d/1vWyV-DukyMIcxinwIhWfnWTTi4XWytdg/view?usp=sharing

[7] Puedes visualizar el diagrama en más detalle en el siguiente enlace: https://drive.google.com/file/d/1SmhteSAG7VeULRekd9EV4eRWxjzHuR9X/view

3.6 MODELO DE DATOS

El modelo de datos contempla las entidades que interoperan en el funcionamiento de la aplicación. Ya que esta memoria es un nuevo módulo de un proyecto ya existente, la mayoría de las tablas habían sido creadas para la versión presentada en la Feria de Software 2018. No obstante, fue necesario realizar algunas modificaciones como agregar una tabla para registrar las patentes en el sistema, así como establecer una relación con el usuario que la ingresa, quedando como llave secundaria de la tabla.

Otra modificación de menor escala consistió en agregar un campo en la tabla de usuarios para manejar los identificadores para hacer envío de notificaciones mediante OneSignal. En la Figura 10 se observa un diagrama con el modelo de datos utilizado por Easy Access.

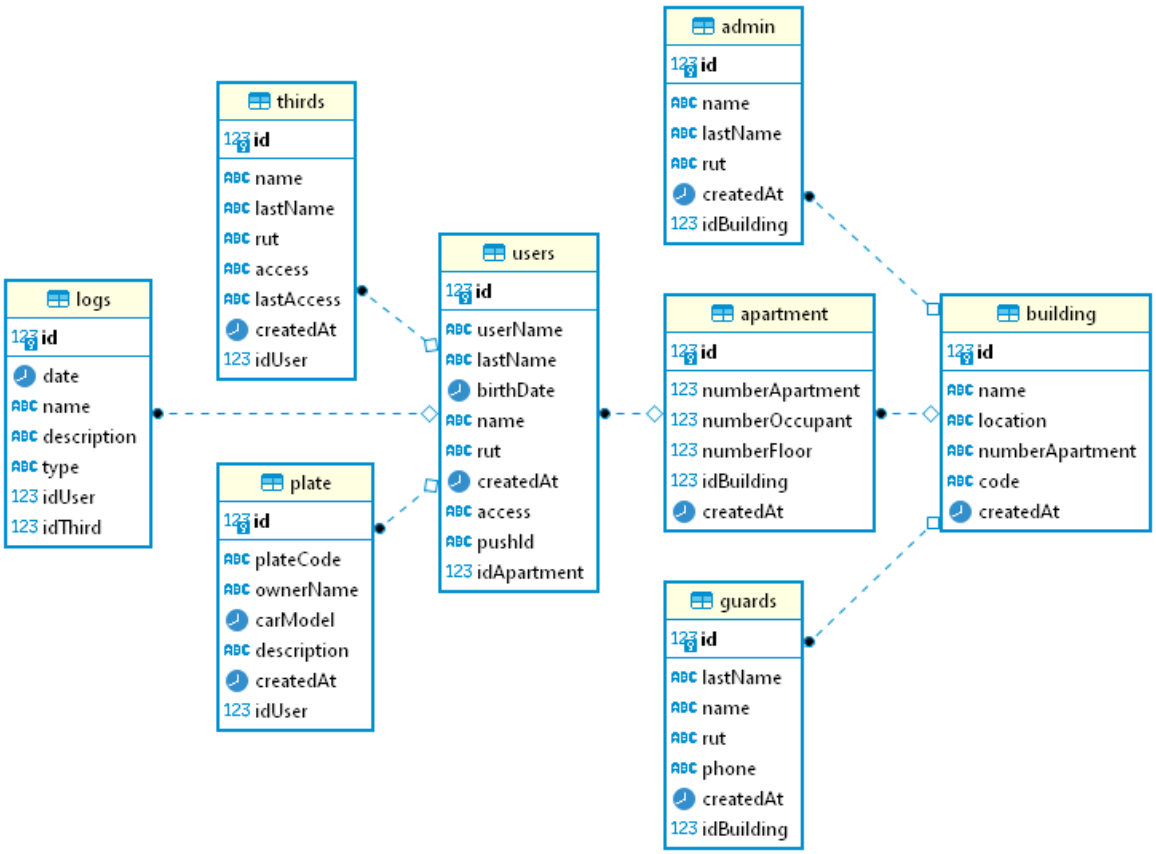

Figura 10: Modelo de datos. Fuente: Elaboración propia.

3.7 DISEÑO DE INTERFAZ DE USUARIO

La interfaz de usuario cuenta con 8 vistas, cada una con diferentes propósitos e interacciones. Fue construida considerando criterios de usabilidad para procurar no saturar al usuario con demasiada información y buscando esta sea tanto legible como de fácil acceso.

Se utilizó iconografías pertenecientes a las bibliotecas del framework y se integró adicionalmente la biblioteca flutter emoji. La incorporación de emojis en la experiencia de usuario permite reforzar o evocar sentimientos en el usuario. En el caso de Easy Access se utilizan para dar *feedback* al usuario respecto de algún paso clave que esté generando cambios en la aplicación.

Se utilizaron mensajes para explicar brevemente el propósito de la interfaz e informar a nuevos usuarios sobre el uso del sistema.

Como referencia se siguieron algunas de las recomendaciones de la tesis de grado en ingeniería multimedia “Análisis y diseño de interfaces centrado en la UX“ (Torregrosa, 2017), en dicho documento el autor desglosa diversos estándares y conceptos de diseño de interfaces de otros autores elaborando finalmente una recomendación sobre ciertos lugares comunes a la hora de construir una experiencia de usuario efectiva.

3.7.1 VISTA ONBOARDING

La vista de onboarding (Figura 11) revela al usuario dos posibles acciones, ingresar a la plataforma o solicitar una cuenta nueva.

Figura 11: Diseño Vista Onboarding. Fuente: Elaboración propia

3.7.2 VISTA LOGIN

La vista de login (Figura 12) presenta dos campos de texto que deben ser rellenados por el usuario, uno es el RUT y el siguiente es la contraseña.

Con esto el usuario es invitado a la acción de ingresar, en caso de poseer la información que solicita el formulario o de recuperar su contraseña en caso de haberla olvidado.

Alternativamente muestra mensajes de error, si es que por algún motivo ingresa información errónea o existe algún problema con el servicio.

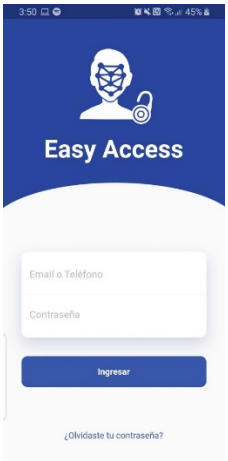

Figura 12: Diseño Vista Login. Fuente: Elaboración propia

3.7.3 VISTA HOME

Esta vista, como se visualiza en la Figura 13, se dividió en 4 secciones, la primera es una barra que muestra el nombre del usuario y su fotografía en caso de poseer una.

La segunda corresponde a información general respecto del número de patentes y personas que ha registrado en la plataforma. Estos fueron diseñados a modo de botón, con el propósito de que el usuario ingrese para gestionar dichas entidades.

Una tercera sección que corresponde a una funcionalidad futura, busca notificar al usuario los gastos comunes, así como también de otros gastos básicos asociados al departamento.

Y finalmente un botón que permite autorizar más patentes o personas a hacer ingreso al edificio utilizando Easy Access.

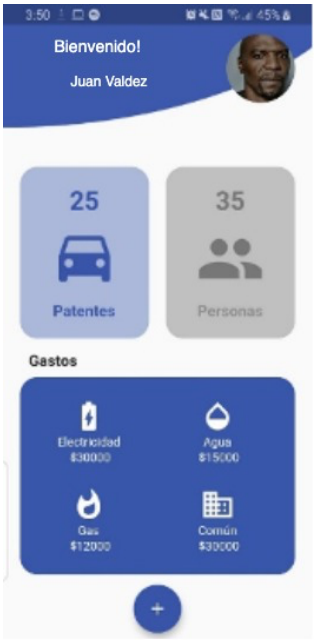

Figura 13: Diseño Vista Home. Fuente: Elaboración propia.

3.7.4 VISTA PATENTES

Esta vista, como se visualiza en Figura 14, consiste en 3 secciones, la primera corresponde a una barra de búsqueda que además presenta un texto explicativo respecto de qué permite realizar la vista.

En la segunda sección, se encuentra un listado de las patentes que actualmente registra el usuario en el sistema con 3 categorías con las que puede filtrar: permitidas, denegadas o todas. Finalmente, un botón que permite agregar nuevas patentes.

Figura 14: Diseño Vista Patentes. Fuente: Elaboración propia.

3.7.5 VISTA AGREGAR NUEVA PATENTE

Esta vista corresponde a un formulario dividido en partes, con el propósito de evitar el desplazamiento vertical. En éste, el usuario puede elegir si desea agregar una persona o patente nueva al sistema.

En la Figura 15, se indica la primera parte del flujo para ingresar una nueva patente invitada. El usuario deberá seleccionar que desea ingresar una patente, presionando el botón marcado en rojo. Una vez hecho esto se presentan dos opciones, agregar patente o agregar persona, recordemos que Easy Access también permite agregar rostros de personas para autorizar su acceso. Luego se encuentran dos campos a rellenar, el primero corresponde al nombre de la persona y, en segundo lugar, hasta cuando se le permitirá el acceso. El usuario debe presionar el botón continuar, una vez que haya rellenado los campos.

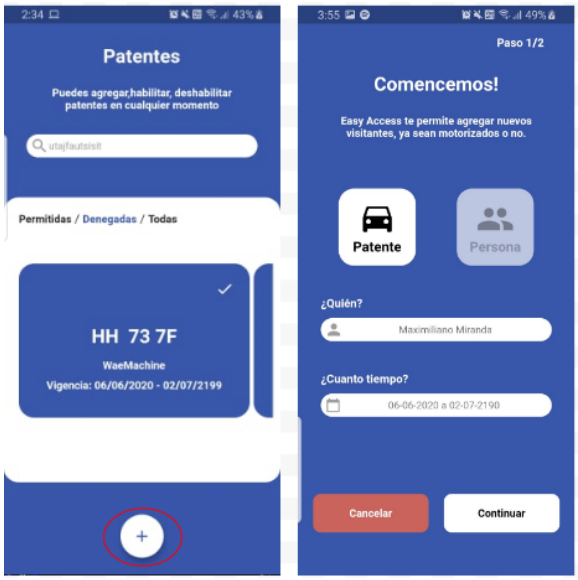

Figura 15: Diseño Stepper parte 1, Flujo nueva patente. Fuente: Elaboración propia

En dicha ventana se le indica al usuario que está por completar el formulario de registro y se le invita a escribir la patente y alguna nota respecto del vehículo para tener en consideración por parte de un usuario administrador. Para preservar los cambios, el usuario debe presionar Finalizar, en ese momento el usuario navega hacia una ventana que indica que el proceso fue concluido con éxito, presentando un botón para volver a la ventana de patentes. Esto se puede observar en la Figura 16**.**

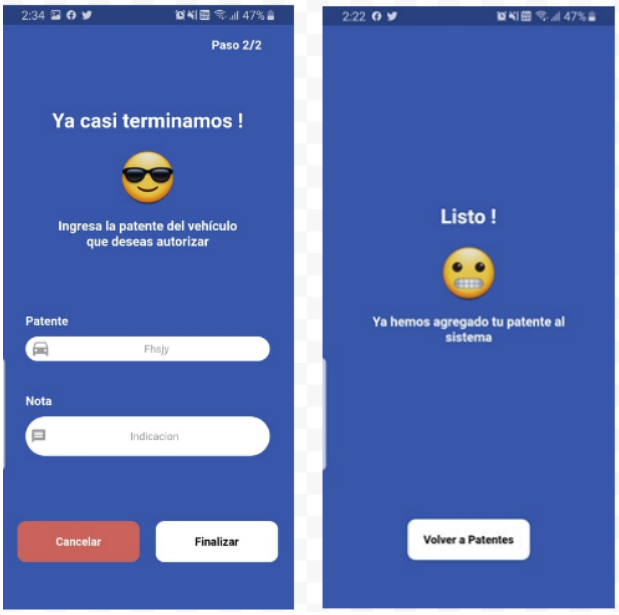

Figura 16: Diseño Stepper parte 2, Flujo nueva patente. Fuente: Elaboración propia

CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN

4.1 IMPLEMENTACIÓN CAPA BACKEND

Como el proyecto había sido desarrollado previamente, este contaba con una API que manejaba alguna de las lógicas que este nuevo módulo requería. Sin embargo, fue necesario agregar y adaptar parte del código existente para soportar el uso de patentes.

Algunas de las modificaciones realizadas incluyen la creación de un nuevo objeto patente, en éste se incorpora la data relevante para el objeto como el código de patente o la fecha en que expira el permiso y una referencia a la tabla de usuarios para identificar a quién pertenece la patente agregada.

También fue necesario agregar un nuevo controlador encargado de gestionar la lógica sobre el objeto patente, se agregaron los métodos de **CRUD** básicos además de búsquedas por criterios personalizados como id de usuario, id de patente o código de patente.

Junto con lo anterior se incorporó para las notificaciones el servicio de OneSignal. Este permite enviar mensajes a los usuarios suscritos, siendo dentro de las opciones gratuitas, la más rápida de implementar además de su versatilidad, ya que posee un SDK para el framework Flutter, Android y Rest. Lo que lo hace el candidato perfecto para este tipo de proyectos multiplataforma.

Se generó una funcionalidad que permite al usuario suscribirse a recibir notificaciones almacenando el Token entre sus datos de usuario.

4.2 IMPLEMENTACIÓN API DE RECONOCIMIENTO DE PATENTE

Se generó una API que encapsula al servicio de Plate Recognizer, con el fin de hacer la integración lo más transparente posible, permitiendo cambiar de servicio sin problemas, en caso de ser requerido.

En ésta se expusieron dos métodos, uno para reconocer imágenes desde una URL y otro para reconocer imágenes en base64.

4.3 IMPLEMENTACIÓN DE INTERFAZ DE USUARIO

La interfaz de usuario se divide entre una aplicación para móviles y un cliente web.

El primero es de cara al usuario, quien habita el departamento, mientras que el segundo es utilizado por entes administrativos, tales como guardias, conserjes o administrador del edificio.

Si bien el proyecto ya contaba con aplicaciones hechas en **Ionic**, se consideró generar una aplicación nueva ya que el estado actual del **framework** en términos de rendimiento no es el mejor. **Flutter** como se indicó en el segundo capítulo fue elegido para dicha tarea.

Para la capa móvil fue necesaria la creación de **modelos** y de **providers** para manejar las llamadas al **backend**.

Los detalles de interfaz fueron abordados en el capítulo anterior. Cabe mencionar se abordó sólo el flujo de placas patentes, por lo que el usuario puede agregar, permitir acceso y ver información respecto de la patente, así como también filtrar la data.

4.4 IMPLEMENTACIÓN DASHBOARD ADMINISTRADOR

Se implementó una sección que permite el listado de todas las patentes en el sistema. Con esto quien administra, puede en caso de ser necesario, editar la información perteneciente a una placa patente.

De momento los campos a editar son: nombre de la persona que fue invitada, código de patente y acceso. Los demás campos corresponden a fechas y a la persona que lo registró, por lo que no constituye un elemento editable.

4.5 DESARROLLO DE PROTOTIPO MÓDULO FÍSICO

El módulo físico cuenta con una cámara y una pantalla para entregar feedback visual y una Raspberry en el que el código vive.

En la Tabla 3 se pueden observar algunos de los componentes electrónicos que se utilizaron en el desarrollo del módulo, junto con el detalle técnico de estos.

Tabla 3: Componentes electrónicos módulo físico. Elaboración Propia.

Componentes electrónicos del módulo de detección	
Artefacto	Características
Raspberry Pi 3 Modelo B+	https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf
LCD Module PI TFT	3.5 pulgadas ; Resolución 320x480
Cámara V2 para respberry	8 megapixeles

Para encapsular el dispositivo se diseñó una carcasa que permite incrustar la Raspberry junto con la pantalla además de la cámara. Además, expone los puertos que tiene la raspberry y permite conectar el transformador.

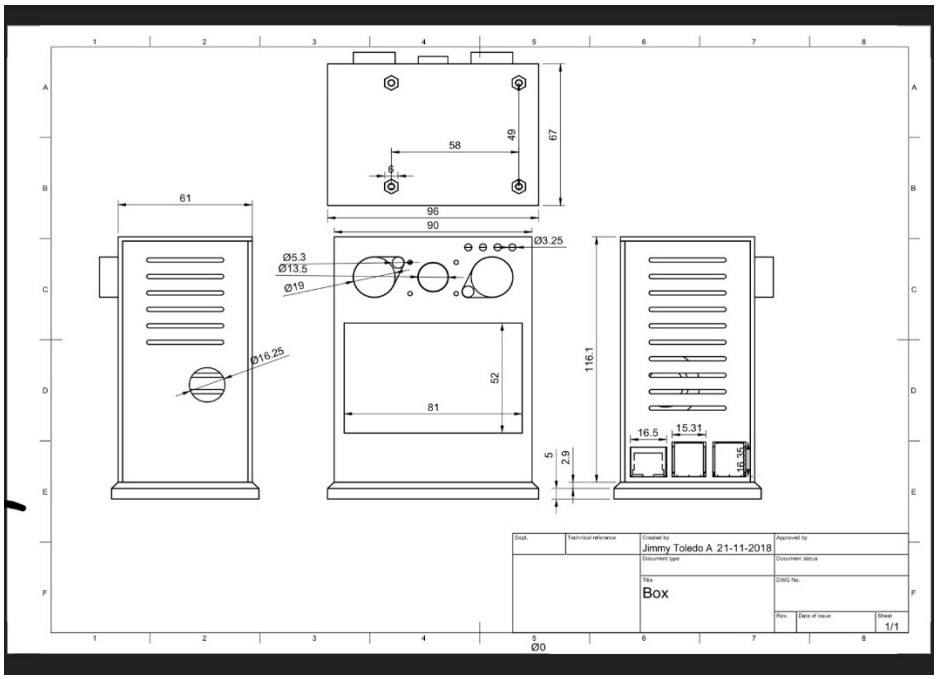

Figura 17 : Esquemático y Render de la manufactura del empaquetado. Elaboración propia.

4.6 INTEGRACIÓN DE MÓDULO FÍSICO AL SISTEMA

Se escribió un pequeño script en Python que sincroniza los componentes en el módulo físico para su correcto funcionamiento. Este programa detecta el pulso sobre el botón, abriendo el dispositivo de captura, codificando la foto y enviando una solicitud a nuestro servicio de reconocimiento de patentes.

CAPÍTULO 5: VALIDACIÓN DE LA SOLUCIÓN

Es importante que al entorno de una aplicación de software, se apliquen pruebas para lograr un efectivo funcionamiento de los artefactos que se construyen.
Existen múltiples maneras de probar una aplicación, las dos categorías más generales en calidad de software son las pruebas funcionales y las no funcionales.
Las no funcionales se encargan de medir aspectos que no se relacionan directamente con la funcionalidad que provee el software. Por ejemplo, medir la tolerancia a fallos que tiene un sistema, la capacidad para resistir múltiples llamadas en simultaneo, entre otras cualidades.
Las pruebas funcionales en cambio, se centran en comprobar aquello que se determinó en los requerimientos al comenzar el proyecto, que el usuario pueda o no realizar determinada acción y que el sistema responda en concordancia.
Para la validación se utilizaron pruebas funcionales en la capa de **frontend** y pruebas no funcionales para la capa de **backend**. En concordancia, se realizó un plan de pruebas basándose en las historias de usuario descritas previamente para probar la aplicación móvil. Mientras que se realizaron test de carga para el **backend**.

5.2 PRUEBAS PERFORMANCE API BACKEND

Pruebas de carga
Las pruebas realizadas sobre la API se hicieron utilizando loader.io, una herramienta web que permite la creación de pruebas de carga utilizando como parámetros el **endpoint** sobre el cual se ejecutarán las pruebas, el tiempo que durará la prueba y el número de clientes ficticios que se utilizarán.
Configuración y descripción de la prueba
En la Tabla 4, se muestran los valores y parámetros solicitados por loader.io para la realización de la prueba, se utilizaron los valores máximos permitidos por la plataforma en su capa gratuita.

Con esta prueba se busca validar el funcionamiento y respuestas de la API bajo condiciones de uso normales, es decir, verificar qué ocurre cuando varios clientes en simultaneo intentan consultarle. El resultado esperado es que el **backend** tolere las llamadas, sin tener problemas o pérdidas de datos. Con un tiempo de respuesta promedio no mayor a 600 ms.

Parámetro	Valor
Clientes	10000
Duración de la prueba	60 segundos
Timeout	10 segundos

Tabla 4: Configuración prueba de carga. Fuente: Elaboración propia.

En la Figura 18 se muestra el resultado de ejecutar la prueba, en específico se entrega el gráfico de tiempo de respuesta vs cantidad de solicitudes de usuario. Como es de esperar, mientras más solicitudes se procesan, más sube la latencia del servicio. Se observa que con cerca de 500 clientes, nuestra aplicación tiene un tiempo de respuesta de 403 ms en promedio. Este número en gran medida se explica por la latencia existente entre nuestro país y el servidor que se aloja en Francia[8].

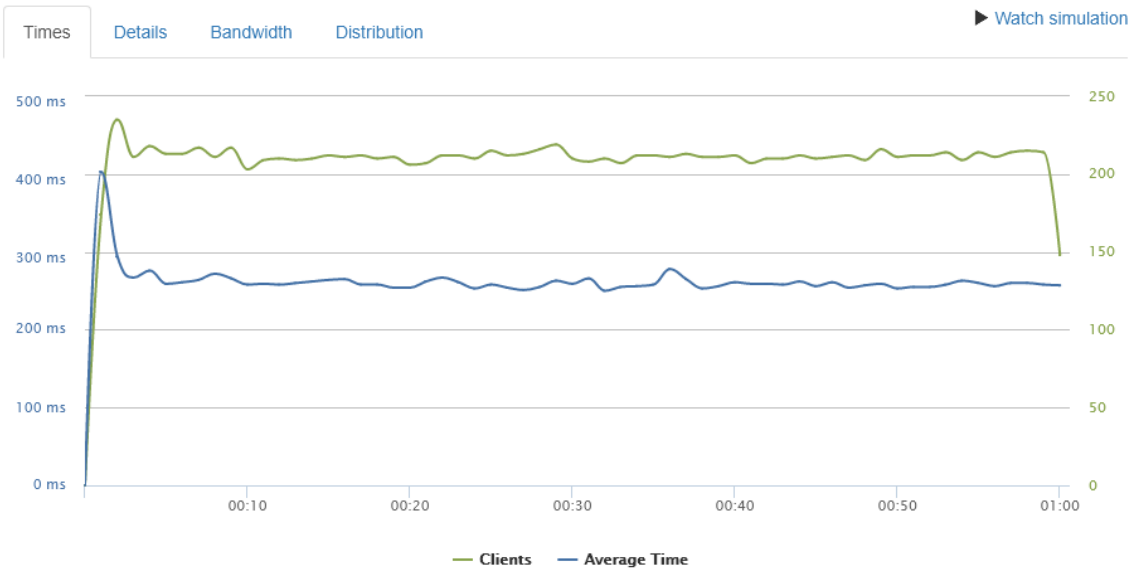

Figura 18: Gráfico de resultados, tiempo de respuesta vs cantidad solicitudes de usuario. Fuente: loader.io.

Además, se observa que el tiempo mínimo de procesamiento de una solicitud realizada por el cliente fue de 208 ms, lo que concuerda con el mínimo **ping** a nuestro servidor producto de la distancia física a la que se encuentra. Esto ocurre con 148 solicitudes por segundo. Lo que es un buen número para nuestra escala. Como referencia Visa procesa hasta 65000[9] transacciones por segundo con una escala de operación que bordea los 61 millones[10] de comercios alrededor del mundo.

De la Figura 19 se puede deducir que no hubo fallos de conexión con el servidor, pues tenemos 0 error por segundo, es decir que el 100% de las solicitudes fue atendida con éxito.

[8] Scaleway se encuentra en Francia teniendo un ping promedio de 200 ms desde América.

[9] https://www.visa.com.ar/acerca-visa/sala-de-noticias/notas-de-prensa/visa-alerts.html

[10] https://usa.visa.com/visa-everywhere/blog/bdp/2020/04/28/supporting-the-worlds-1588044490200.html

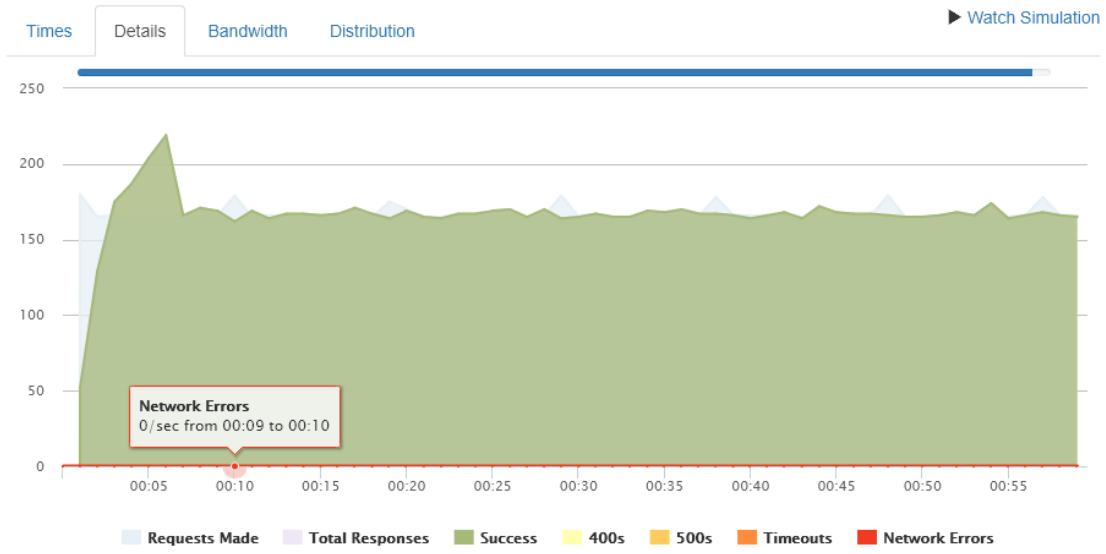

Figura 19: Gráfico de llamadas exitosas, erradas o time out vs tiempo. Fuente: loader.io

Sin embargo, se puede apreciar que hubo un pequeño desface entre las solicitudes realizadas y las atendidas, esto se debe principalmente al aumento de latencia provocado por el aumento en el número de clientes.

5.3 PRUEBAS FUNCIONALES APLICACIÓN MÓVIL

El objetivo de esta sección es recoger los casos de prueba que verifican que el sistema satisface los requerimientos especificados en las historias de usuario. Por lo que contiene la definición de los casos de prueba. En cada prueba se incluyen los requisitos y estrategia a seguir.

Tabla 5: Plan de prueba aplicación móvil. Fuente: Elaboración propia

C1	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Registro Usuario	Prueba de feedback en caso de usar caracteres especiales	- Aplicación móvil instalada
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
	- Prueba nombre usuario sin caracteres especiales	Éxito	Éxito	

C2	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Registro Usuario	Prueba la creación de un nuevo usuario en la aplicación	- Aplicación móvil instalada
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
	- Prueba nombre usuario sin caracteres especiales	Éxito	Éxito	

C3	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Registro Usuario	Prueba de feedback en caso de solo ingresar contraseña	- Aplicación móvil instalada
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
	- Prueba sin ingresar usuario	Advertencia	Error	

C4	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Registro Usuario	Prueba de feedback en caso de solo ingresar el usuario	- Aplicación móvil instalada
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
	- Prueba sin ingresar contraseña	Advertencia	Advertencia	

C5	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Login Usuario	Prueba de feedback en caso de solo ingresar contraseña	- Aplicación móvil instalada - Usuario registrado exitosamente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
	- Prueba sin ingresar usuario	Advertencia	Advertencia	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Login Usuario	Prueba de feedback en caso de solo ingresar el usuario	- Aplicación móvil instalada - Usuario registrado exitosamente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C6	- Prueba sin ingresar contraseña	Advertencia	Advertencia	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Login Usuario	Prueba feedback ingreso de un usuario a la aplicación con contraseña incorrecta	- Aplicación móvil instalada - Usuario registrado exitosamente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C7	- Prueba ingresando contraseña equivocada	Error	Error	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Login Usuario	Prueba el ingreso de usuario la aplicación utilizando datos incorrectos	- Aplicación móvil instalada - Usuario registrado exitosamente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C8	-Prueba ingresando datos correctos	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Login Usuario	Prueba el ingreso de usuario la aplicación utilizando datos incorrectos	- Aplicación móvil instalada - Usuario registrado inválido
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C9	- Prueba ingresando usuario equivocado	Error	Error	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Home	Prueba de interfaz carga datos adecuadamente según usuario	- Aplicación móvil instalada - 2 Usuarios registrados Válido
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C10	- Comparar Datos de usuario	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Home	Prueba interfaz navega correctamente hacia agregar patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C11	- Presionar el botón más debe navegar hacia el selector de agregar patente o agregar usuario	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Home	Prueba interfaz navega correctamente hacia listado de patentes	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C12	- Presionar el botón más debe navegar hacia la ventana de listado de patentes agregadas	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Listado Patente	Prueba interfaz listado de patentes se despliega correctamente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C13	- Navegar sobre el listado de patentes	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Listado Patente	Prueba de filtros listado de patentes	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C14	- Escribir un filtro utilzando una palabra cualquiera	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Listado Patente	Prueba de filtros listado de patentes	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C15	- Escribir un filtro vacio	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Listado Patente	Prueba interfaz navega correctamente hacia agregar patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C16	- Ingreso de datos sin patente	Advertencia	Advertencia	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Registro Patente	Prueba interfaz navega correctamente hacia agregar patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C17	- Ingreso de datos completos	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Registro Patente	Prueba interfaz agregar patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C18	- Ingreso de datos con caracteres especiales.	Advertencia	Advertencia	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Registro Patente	Prueba interfaz agregar patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C19	- No enviar formulario de ingreso, navegar hacia atrás, debería no agregar la patente. Verificar	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Notificaciones	Prueba de feedback al usuario	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
C20	Estrategia	Resultado obtenido	Resultado esperado	Comentarios

	- Seguir flujo de agregar patente, debería notificar al usuario	Éxito	Éxito	
	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Notificaciones	Patente reconocida por el sistema debería enviar notificación al usuario	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C21	- Reconocer patente utilizando módulo físico, el usuario debería ser notificado en caso de que la patente corresponda a una de las que tiene autorizadas	Éxito	Éxito	
	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Editar Patente	Prueba interfaz navega correctamente hacia editar patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado - Seleccionar editar patente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C22	- Desde el listado presionar el botón de editar en la card de patente	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Editar Patente	Prueba de edición de patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado - Seleccionar editar patente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C23	- Ingreso de datos completos. Debería navegar hacia el listado de patentes, la patente debería presentarse modificada	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Editar Patente	Prueba de edición de patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado - Seleccionar editar patente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C24	- Ingreso de datos con caracteres especiales.	Advertencia	Advertencia	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Editar Patente	Prueba de edición de patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado - Seleccionar editar patente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C25	- No enviar formulario de edicion, navegar hacia atrás, debería no agregar la patente -Verificar que no se modificó	Éxito	Éxito	

	Plataforma	Funcionalidad	Descripción	Prerrequisitos
	Aplicación móvil	Editar Patente	Prueba de edición de patente	- Aplicación móvil instalada - Usuario registrado exitosamente - Usuario logeado - Seleccionar editar patente
	Estrategia	Resultado obtenido	Resultado esperado	Comentarios
C26	- Ingreso de datos con vacios	Error	Error	

CAPÍTULO 6: CONCLUSIONES

El objetivo principal de esta memoria, era el de dotar a Easy Access, de un módulo que permitiese el control de acceso a un lugar tan relevante como lo es el estacionamiento. Las técnicas principales empleadas para estos lugares tienden a fallar puesto que dependen en gran medida del factor humano.

Por ejemplo, suele pasar con las puertas que utilizan control remoto que por mal uso se traban. Esto debido principalmente a que los usuarios al no conocer el funcionamiento del sistema, junto con su impaciencia pueden averiarlo al detenerlo a la mitad del flujo de apertura o cerrado. Con Easy Access no requieres de un control que accione tu puerta. Y en ese sentido se cumplió el objetivo.

Analizando en detalle los objetivos específicos podemos encontrar el de diseñar, desarrollar e integrar un API de reconocimiento de placas patentes. Finalmente al realizar una detallada investigación se optó por utilizar un servicio e integrarlo a la plataforma.

Se creó alrededor de la plataforma un servicio web que permita la comunicación con el servicio Plate Recognizer escogido. Por lo que podríamos decir que el objetivo no se cumplió en rigor. Sin embargo esta abstracción permite la creación de un API nueva o la integración de otra con total libertad. Por lo que el sistema es en si agnóstico al API de Plate Recognizer, dado que la aplicación (Easy Access) se comunica con este servicio construido alrededor.

En cuanto a las herramientas utilizadas, se puede decir que fueron bastante acertadas. Las capacidades de Flutter como framework híbrido de desarrollo, sorprenden gratamente al desarrollador.

La calidad de la interfaz, el rendimiento y el lenguaje con el que se codifica (Dart) lo hacen un framework bastante cómodo de aprender y de trabajar. Al ser también una herramienta relativamente vanguardista, conocerla, entenderla y haberla utilizado en un proyecto, entregó valiosos conocimientos de utilidad en la vida profesional.

Por el lado del backend, NodeJs es un framework cada vez más demandado en la industria. El manejo de concurrencia y el amplio soporte de la comunidad que día a día crea nuevos plugins, le da una gran validez como herramienta.

En el caso de este proyecto en particular, fue una buena opción principalmente porque es lo suficientemente flexible para abordar casos simples como CRUD e integraciones más complejas como lo fue el del API de Plate Recognizer.

Respecto del servidor utilizado, ScaleWay es un buen espacio para realizar pruebas de concepto y proyectos pequeños. Se recomienda su uso solo para prototipado, ambiente de testing o QA. Para ambientes productivos es recomendable utilizar alguna plataforma más robusta como GCP(Google Cloud Platform).

Principalmente por la confianza que entrega y las numerosas herramientas que hacen del despliegue una actividad más segura. Por ejemplo, Cloud Run maneja versiones de los despliegues realizados hacia el contenedor, pudiendo seleccionar versiones anteriores para dirigir el tráfico de los usuarios. Algo que puede resultar muy útil cuando se está en producción y ocurre una falla producto de código recientemente incluido o de casos de borde que pasaron los testeos previos. Cosas que ScaleWay no tiene.

Por otro lado, ScaleWay tiene algunos problemas muy recurrentes de conectividad, esto debido principalmente a la infraestructura física de la compañía, que precisamente para abaratar costos utiliza recursos de menor calidad.

Respecto de la metodología, considero que Scrum es altamente efectivo para la gestión de tareas en equipos y que se beneficia de las iteraciones. Respecto a las metodologías de desarrollo, se pudo haber empleado por ejemplo TDD (Test Driven Development) para complementar el aseguramiento de la calidad.

Con TDD primero se crea la prueba y luego el código que permite pasar la prueba, lo que te genera la posibilidad armar un flujo de desarrollo con prácticas devops (CI/CD), continous integration y delivery. Devops agiliza el desarrollo y disminuye el error, mejorando la calidad del software construido.

Respecto de la metodología de escritura, se pudo haber empleado algún sistema de manejo de tiempos como Pomodoro. Esta técnica consiste en determinar un tiempo fijo y cronometrar su transcurso. Al terminar dicho bloque, de debe descansar una cantidad proporcional al tiempo en que se estuvo trabajando. Esto con el fin de mejorar la concentración y el tiempo utilizado en la construcción de este escrito.

Hubiera cambiado mi enfoque al desarrollar este trabajo. Habría comenzado por redactar el trabajo escrito en vez de solo enfocarme en el aspecto técnico. Haber desarrollado lo técnico primero, trajo algunas consecuencias como una falsa sensación de que ya había completado mi trabajo.

Por otro lado, poseer las definiciones técnicas previamente, hubiera tenido como consecuencia una mayor fluidez al desarrollar. Dado que de antemano se sabe qué tablas crear, qué artefactos construir y finalmente, qué integraciones se hubieran tenido que realizar.

Gracias al API de Plate Recognizer, la solución puede ser fácilmente implementada en más de 90 países. Esto es un punto crítico, ya que se tiene como propósito poder ser implementado en otros mercados a futuro, contar con dicha tecnología permite escalar el servicio y extenderlo con facilidad, sobre todo en países como la India que posee un sistema bastante especial de patentes.

Otro antecedente a destacar es que producto del Coronavirus y el miedo subyacente a utilizar transporte público, ha implicado un aumento en la demanda de vehículos. Algunas redes de concesionarios del país han incrementado sus ventas hasta en 40%. Debido en parte a la liquidez que proporcionaron los retiros de AFP del 10%.

Esto trae consigo desafíos indudablemente. Y es ahí donde el producto desarrollado puede ayudar. Por ejemplo, a recibir de manera autónoma vehículos como invitados. Llevar el control sobre quién entra y sale de un edificio.

Indudablemente, el uso de aplicaciones de control y gestión de la información puede llegar a levantar susceptibilidades. Ha habido casos reconocidos a nivel mundial por vender información que nos pertenece como usuarios. Acá se abre un desafío.

¿Cómo entregar confianza de que la información a utilizar no va a ser vendida o mal utilizada?.

¿Cómo asegurar a los clientes de que solo se utilizará su información para abrir y cerrar el portón y no para que le aparezca publicidad dirigida?. Se debe avanzar en políticas públicas respecto a la privacidad de los datos.

Por otro lado, existen desafíos técnicos propios de la solución implementada. Por ejemplo, detectar el uso inapropiado del sistema. Si es que un vehículo interviene la patente, ya sea colocando un papel o falsificando la patente.

Quizás incorporar un sistema de doble chequeo en donde se analice por una parte la patente, validando su integridad, y por otro, que un humano autorice el ingreso con una captura en tiempo real en video de quien intente hacer ingreso a modo de tener la seguridad de que no están intentando hacer un ingreso indebido.

Otra arista a mejorar o complementar el producto, sería la de extender la solución para también informar sobre la disponibilidad de estacionamientos.

En términos laborales, me aportó en conocimientos sobre tecnologías de vanguardia. Tuve la posibilidad de aprender Flutter y NodeJs. Ambos frameworks altamente demandados en la industria.

Actualmente me desempeño como Desarrollador FullStack, por lo que el haber aprendido tanto de metodologías agiles, desarrollo Backend y FrontEnd fue en parte de porque conseguí dicho trabajo.

La carrera me entregó las base con las que he podido desempeñarme como profesional. Son pocas las veces que no he podido enfrentar un desafío laboral y personal de forma airosa, gracias a la formación que recibí durante mi paso por la universidad.

La confianza en mi mismo, en mis capacidades y el haber enfrentado una cuantiosa cantidad de desafíos, me han definido como profesional.

Bases de Datos y la Feria de Software fueron experiencias clave dentro de mi formación y que fomentaron mis habilidades de trabajo en equipo, liderazgo a la par también con la capacidad técnica.

Taller de Sistemas de Computación, fue un gran aporte en este trabajo. En dicha asignatura se abordaron algunos temas interesantes como la integración de APIs, contenedores de Docker, herramientas de control de versiones (git) y frameworks para el desarrollo de tanto aplicaciones móviles como de web.

Por otro lado, el mayor impacto y formación como profesional lo recibí durante mi práctica en Lifeware, y por consecuencia también se ve reflejada dicha experiencia en el desarrollo de este trabajo.

REFERENCIAS BIBLIOGRÁFICAS

INE. (2019). *Boletín: Sectores Económicos, permisos de edificación autorizada, Edición N°244.* Santiago de Chile: INE.

Torregrosa, R. Y. (2017). *Análisis y diseño de interfaces centrado en la UX.* Alicante: Universidad de Alicante.

Lindahl, T. (2007). *Study of Local Binary Patterns.* Norrköping: Institutionen for teknik och naturvetenskap.

Parking, G. (2012). *Sitio web*. Obtenido de Descripción del producto: www.gestionparking.cl

identification, N. (2010). *Access V2 short-range datasheet*. Obtenido de Sitio web: https://portal.nedapidentification.com/download/ANPR/Datasheet/English/ANPR

identification, N. (2010). *Producto moov software*. Obtenido de Sitio web: https://www.nedapidentification.com/es/productos/moov/moov-software/

identification, N. (2012). *Access V2 short-range datasheet*. Obtenido de Sitio web: https://portal.nedapidentification.com/download/ANPR/Datasheet/English/ANPR

identification, N. (2012). *producto anpr lumo*. Obtenido de Sitio web: https://www.nedapidentification.com/es/productos/anpr/anpr-lumo/

Guerrero, A. S. (2017). *Análisis de capacidades, diferencias, roadmap y futuras posibilidades de Ionic, React Native and NativeScript*. Obtenido de Mimacom: https://blog-es.mimacom.com/roadmap-ionic-react-ns/

Rosebrock, A. (2020). *OpenCV: Automatic License/Number Plate Recognition (ANPR) with Python*. Obtenido de pyimagesearch: https://www.pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-anpr-with-python/

Rosebrock, A. (2015). *Local Binary Patterns with Python & OpenCV*. Obtenido de pyimagesearch: https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/

Sable, F. Z. (2021). *A comprehensive guide to OCR with Tesseract, OpenCV and Python*. Obtenido de nanonets: https://nanonets.com/blog/ocr-with-tesseract/

Dahiya, P. K. (2017). Localization Techniques in ANPR Systems: A-State-of-Art. *Intelligent Transportation System*.