

# Sistema automático de verificación de cumplimiento de planogramas usando inteligencia artificial aplicada a imágenes

Albacar Fernández, Pol

Curs 2018-2019

Director: Horacio Saggion

GRAU EN ENGINYERIA DE SISTEMES AUDIOVISUALS



Universitat  
Pompeu Fabra  
Barcelona

Escola  
Superior Politècnica

**Treball de Fi de Grau**

Sistema automático de verificación de cumplimiento de  
planogramas usando inteligencia artificial aplicada a  
imágenes

Pol Albacar Fernández

---

TRABAJO FINAL DE GRADO  
GRADO EN INGENIERÍA EN SISTEMAS AUDIOVISUALES  
ESCUELA SUPERIOR POLITÉCNICA UPF  
AÑO 2019

DIRECTOR DEL TRABAJO

Horacio Saggion, Natural Language Processing Research Group





A mis padres, por el apoyo incondicional.



## **Agradecimientos**

Primeramente, quiero agradecer a mis compañeros de Pervasive Technologies, no podría haber encontrado una empresa mejor en la que realizar prácticas. Muchas gracias por ofrecerme la posibilidad de realizar este proyecto, por el conocimiento que me habéis proporcionado y por la pasión que le ponéis al día a día.

Gracias también a todas las personas que han formado parte de mi etapa universitaria, desde los profesores que dedican sus esfuerzos a nuestro aprendizaje, hasta mis compañeros sin los cuales esta experiencia no hubiera sido tan gratificante.

Finalmente, agradezco a mis padres Felicidad Fernandez y Fernando Albacar, a mi hermana Judith y a Clàudia. Gracias por haberme apoyado en todo momento y hacer que esto sea posible.



## **Resumen**

El proyecto consiste en desarrollar un sistema ofrecido en forma Web App apoyado por una arquitectura de sistemas robusta con la principal funcionalidad de ayudar a las compañías a conseguir un control completo del cumplimiento de los planogramas de sus productos en los diferentes puntos de venta. El sistema utiliza técnicas de reconocimiento automático de objetos en imágenes mediante el uso de tecnologías de inteligencia artificial. Estas tecnologías se aplican a las imágenes que capturan los gestores de punto de venta en los diferentes comercios que visitan y así proporcionar un análisis objetivo y rápido de la visibilidad de los productos. El análisis consiste en una comparación entre la visibilidad de los productos contratada por la marca y la visibilidad que estos realmente tienen, con el objetivo de proporcionar una evaluación y extracción de datos útiles.

## **Abstract**

The project is focused on developing a system offered in a Web App and supported by a robust system architecture with the main functionality of helping companies to achieve complete control of the compliance of the planograms of their products in the different points of sale. The system uses techniques of automatic recognition of objects in images through the use of artificial intelligence technologies. These technologies are applied to the images captured by the point of sale managers in the different stores they visit and thus provide an objective and fast analysis of the visibility of the products. The analysis consists of a comparison between the visibility of the products contracted by the brand and the visibility they really have, with the aim of providing an evaluation and extraction of useful data.



# Índice

|  |             |
|--|-------------|
| <b>Resumen .....</b>   | <b>vii</b>  |
| <b>Lista de figuras .....</b>  | <b>xiii</b> |
| <b>Lista de tablas .....</b>   | <b>xv</b>   |
| <b>1. INTRODUCCIÓN .....</b>   | <b>1</b>    |
| 1.1 Introducción al problema.....                                      | 1           |
| 1.2 Definición de conceptos del sector.....                            | 2           |
| 1.2.1 Planograma .....   | 2           |
| 1.2.2 Realograma.....  | 3           |
| 1.2.3 Lineal.....  | 3           |
| 1.2.4 Expositor.....   | 3           |
| 1.2.5 Frentes de exhibición o facing.....                              | 4           |
| 1.2.6 Gestores de punto de venta (GPV) .....                           | 4           |
| 1.2.7 Trade marketing.....   | 5           |
| 1.2.8 SKU .....  | 5           |
| 1.3 Estado del arte .....  | 5           |
| 1.4 Requerimientos y funcionalidades .....                             | 6           |
| 1.4.1 Requisitos funcionales para gestores de puntos de venta .....    | 6           |
| 1.4.2 Requisitos funcionales para responsables de Trade Marketing..... | 6           |
| 1.5 Tecnología empleada.....   | 9           |
| 1.5.1 Google Cloud Platform.....                                       | 9           |
| 1.5.2 Diagrama de la arquitectura completa del sistema .....           | 10          |
| <b>2. PIPELINE PREDICTIVO.....</b>                                     | <b>13</b>   |
| 2.1 Google Cloud Storage .....   | 13          |
| 2.1.1 Google Cloud Storage .....                                       | 13          |
| 2.1.2 Google Cloud Storage aplicado al proyecto .....                  | 14          |
| 2.2 Google Cloud Firestore .....                                       | 15          |
| 2.2.1 Cloud Firestore .....  | 15          |
| 2.2.2 Google Cloud Firestore aplicado al proyecto .....                | 16          |
| 2.3 Google Cloud Functions.....  | 19          |
| 2.3.1 Google Cloud Functions.....                                      | 19          |
| 2.3.2 Google Cloud Functions aplicado al proyecto .....                | 20          |
| 2.4 Google Cloud AutoML Vision Object Detection.....                   | 20          |
| 2.4.1 Google Cloud AutoML.....   | 20          |
| 2.4.2 Google Cloud AutoML aplicado al proyecto .....                   | 21          |
| 2.5 Diagrama de la arquitectura del Pipeline Predictivo .....          | 22          |

|   |           |
|---|-----------|
| <b>3. DETECCIÓN DE OBJETOS</b> .....                            | <b>23</b> |
| 3.1 Creación del primer conjunto de datos.....                  | 23        |
| 3.1.1 Análisis de los productos de la marca L'OR.....           | 23        |
| 3.1.2 Captura de imágenes.....                                  | 26        |
| 3.1.3 Etiquetado.....   | 27        |
| 3.1.4 Subir las imágenes a Google Cloud Storage.....            | 32        |
| 3.1.5 Transformación de los datos para adaptarlos a AutoML..... | 33        |
| 3.1.6 Importar el archivo CSV a AutoML.....                     | 34        |
| 3.1.7 Eliminar etiquetas escasas.....                           | 35        |
| 3.2 Primer entrenamiento.....                                   | 36        |
| 3.3 Evaluación del primer modelo.....                           | 37        |
| 3.4 Creación del segundo conjunto de datos.....                 | 40        |
| 3.5 Segundo entrenamiento.....                                  | 41        |
| 3.6 Evaluación del segundo modelo.....                          | 41        |
| 3.7 Comparación entre ambos modelos.....                        | 44        |
| <b>4. DESARROLLO DE LA APLICACIÓN</b> .....                     | <b>47</b> |
| 4.1 Diseño.....   | 47        |
| 4.1.1 Material UI.....  | 47        |
| 4.1.2 Descripción por pantallas.....                            | 47        |
| 4.1.3 Mapas de navegación.....                                  | 64        |
| 4.1.4 Avisos de errores y éxitos.....                           | 65        |
| 4.2 Cliente-Servidor.....                                       | 66        |
| 4.3 Parte Cliente.....  | 67        |
| 4.3.1 HTML.....   | 67        |
| 4.3.2 CSS.....  | 67        |
| 4.3.3 Javascript.....   | 67        |
| 4.3.4 ReactJS.....  | 68        |
| 4.3.5 Librerías.....  | 68        |
| 4.4 Parte Servidor.....   | 70        |
| 4.5 APIs & Services.....  | 70        |
| 4.6 Google App Engine.....                                      | 71        |
| 4.7 Cloud Firestore.....  | 72        |
| 4.8 Cloud Storage.....  | 73        |
| 4.9 Diagrama de la arquitectura de la aplicación.....           | 73        |
| <b>5. CONCLUSIONES</b> .....                                    | <b>75</b> |
| 5.1 ¿Qué se ha conseguido?.....                                 | 75        |
| 5.2 Trabajo futuro.....   | 76        |

|                          |           |
|--------------------------|-----------|
| <b>Bibliografía.....</b> | <b>77</b> |
| <b>Anexo 1 .....</b>     | <b>79</b> |



## Lista de figuras

|   |    |
|---|----|
| Figura 1.1: Ejemplo de planograma pictórico con productos de la marca L'OR.....     | 2  |
| Figura 1.2: Muestra de lineal vacío. ....   | 3  |
| Figura 1.3: Ejemplo de un expositor. ....   | 4  |
| Figura 1.4: Diagrama de la arquitectura completa del sistema.....                   | 11 |
| Figura 2.1: Icono de Google Cloud Storage. ....                                     | 13 |
| Figura 2.2: Icono de Google Cloud Storage. ....                                     | 15 |
| Figura 2.3: Icono de Google Cloud Functions. ....                                   | 19 |
| Figura 2.4: Esquema de funcionamiento de las Cloud Functions de Google.....         | 19 |
| Figura 2.5: Icono de Google Cloud AutoML. ....                                      | 21 |
| Figura 2.6: Esquema de funcionamiento de AutoML. ....                               | 21 |
| Figura 2.7: Diagrama de la arquitectura del Pipeline Predictivo.....                | 22 |
| Figura 3.1: Ejemplo de paquetes de café L'OR de 10 cápsulas. ....                   | 25 |
| De izquierda a derecha: RISTRETTO, ONYX y SONTUOSO.....                             | 25 |
| Figura 3.2: Ejemplo de paquetes de café L'OR de 20 cápsulas. ....                   | 25 |
| De izquierda a derecha: RISTRETTO, ONYX y SONTUOSO.....                             | 25 |
| Figura 3.3: Ejemplo de paquetes de café L'OR de 40 cápsulas. ....                   | 25 |
| De izquierda a derecha: RISTRETTO y ONYX. ....                                      | 25 |
| Figura 3.4: Cara frontal (verde), lateral (azul) y superior (roja) en.....          | 26 |
| cajas L'OR de 10 y 20 cápsulas.....   | 26 |
| Figura 3.5: Imágenes de expositores que se han podido realizar frontalmente. ....   | 27 |
| Figura 3.6: Imágenes de expositores que no se han podido realizar frontalmente..... | 27 |
| Figura 3.7: Interfaz del programa LabelImg .....                                    | 29 |
| Figura 3.8: Ejemplo de archivo XML que retorna el software LabelImg. ....           | 29 |
| Figura 3.9: Ejemplos de etiquetado. ....  | 32 |
| Figura 3.10: Evaluación de AutoML del primer modelo. ....                           | 37 |
| Figura 3.11: Imágenes originales y predicciones del primer modelo.....              | 38 |
| Figura 3.12: Imagen con error de detección y clasificación.....                     | 39 |
| Figura 3.13: Error de detección en la imagen de la Figura 3.12.....                 | 39 |
| Figura 3.14: Error de clasificación en la imagen de la Figura 3.12.....             | 40 |
| Figura 3.15: Evaluación de AutoML del segundo modelo.....                           | 42 |
| Figura 3.16: Imágenes originales y predicciones del segundo modelo. ....            | 43 |
| Figura 3.17: Resultados del primer (izquierda) y segundo (derecha) modelo.....      | 45 |
| Figura 3.18: Resultados del primer (izquierda) y segundo (derecha) modelo.....      | 45 |
| Figura 4.1: Pantalla A. Iniciar sesión. ....  | 48 |
| Figura 4.2: Pantalla B. Histórico de visitas. ....                                  | 49 |

|  |    |
|--|----|
| Figura 4.3: Pantalla C. Imagen de la visita.....                               | 50 |
| Figura 4.4: Pantalla D. Detalles de la visita. ....                            | 51 |
| Figura 4.5: Pantalla E. Añadir comentario a la visita.....                     | 52 |
| Figura 4.6: Pantalla F. Puntos de venta. ....                                  | 53 |
| Figura 4.7: Pantalla G. Añadir nuevo punto de venta. ....                      | 54 |
| Figura 4.8: Pantalla H. Punto de venta (1). ....                               | 55 |
| Figura 4.9: Pantalla H. Punto de venta (2). ....                               | 55 |
| Figura 4.10: Pantalla I. Enviar correo electrónico al punto de venta. ....     | 57 |
| Figura 4.11: Pantalla J. Nueva visita, datos del punto de venta. ....          | 58 |
| Figura 4.12: Pantalla K. Nueva visita, fecha y hora de la visita.....          | 59 |
| Figura 4.13: Pantalla L. Nueva visita, selección de la fecha de la visita..... | 60 |
| Figura 4.14: Pantalla M. Nueva visita, selección de la hora de la visita. .... | 61 |
| Figura 4.15: Pantalla N. Nueva visita, subir imagen. ....                      | 62 |
| Figura 4.16: Pantalla O. Gestión de usuarios. ....                             | 63 |
| Figura 4.17: Mapa de navegación entre las pantallas principales. ....          | 64 |
| Figura 4.18: Mapa de navegación desde la pantalla Histórico de visitas.....    | 64 |
| Figura 4.19: Mapa de navegación desde la pantalla Puntos de venta.....         | 65 |
| Figura 4.20: Barra de progreso. ....   | 66 |
| Figura 4.21: Círculo de progreso.....  | 66 |
| Figura 4.22: Snackbar de error. ....   | 66 |
| Figura 4.23: Snackbar de éxito.....  | 66 |
| Figura 4.24: Diagrama de la arquitectura de la aplicación.....                 | 74 |
| Figura A.1: Área común entre dos bounding boxes y área de la unión.....        | 80 |

## Lista de tablas

|   |    |
|---|----|
| Tabla 1.1: Ejemplo de los resultados obtenidos tras una visita.....                     | 8  |
| Tabla 1.2: Cumplimiento individual de cada variedad de productos. ....                  | 8  |
| Tabla 2.1: Formato de los documentos de la colección “usuarios”.....                    | 16 |
| Tabla 2.2: Formato de los documentos de la colección “establecimientos”.....            | 17 |
| Tabla 2.3: Campos añadidos a la colección “establecimientos”. ....                      | 17 |
| Tabla 2.4: Formato de los documentos de la colección “registros”. ....                  | 18 |
| Tabla 2.5: Formato de los documentos de la colección “productos”. ....                  | 18 |
| Tabla 3.1: SKUs de la marca L’OR disponibles en la página web. ....                     | 24 |
| Tabla 3.2: Distribución entre “TRAIN”, “VALIDATION” y “TEST”.....                       | 36 |
| Tabla 3.3: Precisión média, precisión y sensibilidad para el primer y segundo modelo.44 |    |
| Tabla 4.1: Clases de máquinas disponibles en Google App Engine. ....                    | 72 |



# 1. INTRODUCCIÓN

## 1.1 Introducción al problema

En la actualidad, las empresas fabricantes que distribuyen sus productos a grandes superficies y otros establecimientos comerciales de terceros, invierten mucho esfuerzo y recursos en asegurar mínimos de visibilidad de sus productos en los expositores o lineales de los puntos de venta. El objetivo de estas acciones es optimizar las ventas, ya que una de las claves del éxito es la buena visibilidad de los productos, tanto en número de unidades expuestas, mix de productos y situación física en el lineal o expositor. Todos estos factores pueden representar la diferencia entre realizar o perder una venta.

Por esta razón, las empresas (generalmente de gran consumo) invierten muchos esfuerzos en comprobar el cumplimiento de la visibilidad y posicionamiento de sus productos contratados en la negociación en cada uno de los puntos de venta. Este control se realiza mediante los gestores de punto de venta (o GPVs, tal y como se les conoce en el sector del Trade Marketing), empleados del fabricante que se encargan de visitar los diferentes comercios para verificar y generar informes de visibilidad de los productos de la empresa, entre otras funciones. Estos informes, además del tiempo necesario para completarlos, pueden verse afectados por errores de interpretación o de contabilización de los productos por parte del GPV. El gran volumen de informes generados provoca que acabe resultando imposible revisar cada uno de ellos por el departamento de Trade Marketing y, adicionalmente, no se aprovechan todos los datos recogidos por los GPVs.

En este trabajo de final de grado se presenta un sistema innovador para conseguir un análisis completo de la verificación del cumplimiento de planogramas en comercios, utilizando las últimas tecnologías de desarrollo de aplicaciones web, inteligencia artificial y cloud computing. El sistema está basado en una plataforma web, en la que los GPVs pueden cargar imágenes del lineal o expositor del punto de venta visitado y, automáticamente, se realiza un análisis de la imagen para extraer datos cuantitativos y cualitativos de los productos expuestos. Como resultado se genera una evaluación completa de la visibilidad de los productos de la imagen (realograma), comparándola con la visibilidad que se había pactado inicialmente en ese punto de venta (planograma).

Los principales beneficios de este sistema son el obtener resultados robustos a errores, con mucha rapidez, sin la necesidad de verificar manualmente la información obtenida por los GPVs y la posibilidad de extraer varios tipos de indicadores referentes a los productos y su visibilidad que se pueden cruzar con las ventas y el nivel de desempeño del comercio.

El sistema desarrollado en este trabajo ha sido entrenado para realizar una verificación del cumplimiento de planogramas para productos de la marca café L'OR. La marca L'OR, creada por la empresa Jacobs Douwe Egberts (JDE), comercializa cápsulas de café compatibles con las máquinas Nespresso. Sin embargo, el sistema que se desarrolla es completamente extrapolable a cualquier otro tipo de marca o producto.

## 1.2 Definición de conceptos del sector

Con el objetivo de poder hacer un seguimiento más fluido del trabajo expuesto en este documento, se ha añadido esta sección para definir y clarificar conceptos del sector al que se dirige el sistema propuesto y que no son muy comunes en el vocabulario de las personas externas a este.

### 1.2.1 Planograma

Los planogramas son herramientas gráficas de gestión del espacio utilizadas para optimizar, organizar y analizar los espacios de exhibición de los productos en un comercio. En ellos se indican las posiciones y caras de exhibición de los distintos productos que conforman una determinada categoría con la finalidad de ubicarlos de forma que sea rentable, óptima y visualmente atractiva para el consumidor. Los planogramas, que también podemos definir como instrumentos de Trade Marketing, surgen debido a la necesidad de gestionar eficazmente el espacio de exposición en los puntos de venta ante un mercado tan complejo y competitivo como el actual (Boubeta, 2006).

Los planogramas habitualmente se representan utilizando diagramas o croquis en los cuales se describe con detalle el espacio en el que debe ser ubicado cada producto de una estantería, mostrando no solo la posición de los artículos, sino también la cantidad, los frentes de exhibición, altura y la profundidad que deben ocupar. La complejidad de un planograma depende del tamaño del comercio y del espacio disponible en el lineal o expositor.

En la actualidad existen programas informáticos gráficos con los que es posible crear planogramas fácilmente. Las tres posibles representaciones de un planograma que existen son las casillas con texto, los planogramas pictóricos (ejemplificado en la Figura 1.1) y los tridimensionales.



*Figura 1.1: Ejemplo de planograma pictórico con productos de la marca L'OR.*

## 1.2.2 Realograma

El concepto de realograma corresponde a la representación o situación real de los productos en el espacio de exhibición.

## 1.2.3 Lineal

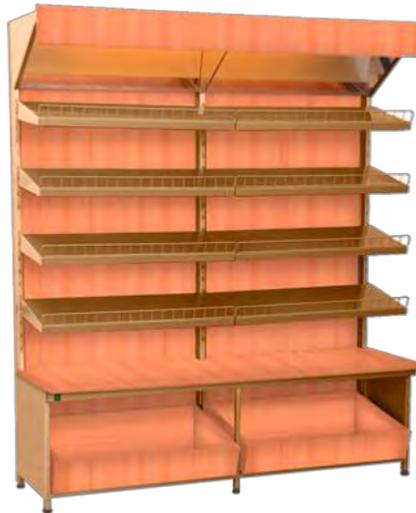
El lineal es un conjunto de estanterías y espacio en el punto de venta en donde los comercios exponen y presentan los productos (Datakey, 2014). El espacio conocido como lineal lo componen las estanterías, vitrinas y cámaras de productos congelados, entre otros. En la Figura 1.2 se muestra un lineal vacío, el mismo que se ha utilizado para la construcción del planograma mostrado en la Figura 1.1.



*Figura 1.2: Muestra de lineal vacío.*

## 1.2.4 Expositor

Mueble en el cual se exponen los distintos productos a la vista del público para su venta pero que, a diferencia del lineal, no forman parte del mobiliario básico del comercio. Las dimensiones de los expositores pueden llegar a ser muy variadas, dependiendo del espacio disponible o de los productos a exhibir en ellos. Forman parte del término conocido como publicidad en el punto de venta. Se muestra un ejemplo de expositor en la Figura 1.3.



*Figura 1.3: Ejemplo de un expositor. Fuente: <http://www.hostelpa.com/>*

### 1.2.5 Frentes de exhibición o facing

El facing, sinónimo de cara o frontal de un producto, es la unidad visible de frente por el cliente en la primera fila de exposición de un lineal o expositor (Kluwer, Wolters). El número de facings varía en función del espacio disponible en el lineal o expositor, del tipo de producto, de las dimensiones y de la cantidad que se ha contratado en un punto de venta.

### 1.2.6 Gestores de punto de venta (GPV)

Los gestores de punto de venta son profesionales que tienen como principal objetivo verificar la situación de los productos de la marca para la cual trabajan y supervisar e implementar las políticas comerciales acordadas en cada punto de venta. Por tanto, las principales funciones del gestor de punto de venta, de acuerdo con (Metafase), son las siguientes:

- ❖ Verificar que los acuerdos de planogramas negociados con las centrales de compra se cumplen en cada uno de los puntos de venta.
- ❖ Garantizar que cada punto de venta mantenga el surtido acordado.
- ❖ Implantar nuevo surtido.
- ❖ Mejorar la presencia y visibilidad de los productos.
- ❖ Gestionar el material en el punto de venta.
- ❖ Presentar a los puntos de venta la gama de productos que la marca tiene en stock central.
- ❖ Generar pedidos de reposición.
- ❖ Mejorar la venta promedio.

### 1.2.7 Trade marketing

Trade Marketing es el departamento de una empresa encargado de establecer diferentes planes de acción y estrategias para transformar el punto de venta, haciéndolo más atractivo para los consumidores con el fin de mejorar su experiencia de compra (InboundCycle, Equipo, 2016). Este tipo de marketing se aplica también a los diferentes canales de distribución de los productos para obtener una mejor salida comercial.

Cada día el nivel de competitividad entre las diferentes marcas en los canales de distribución aumenta, y cada vez es más difícil fidelizar al consumidor final debido, en gran parte, a la gran cantidad y diversificación de productos. Por esta razón, las personas encargadas del Trade Marketing tienen como responsabilidad mantener y mejorar la rotación y visibilidad de los productos en los puntos de venta.

### 1.2.8 SKU

El SKU, sigla que proviene de las palabras inglesas Stock-Keeping Unit, corresponde a un código único usado internamente por las empresas para identificar a una misma variedad de productos, conocido también como código de referencia del producto.

## 1.3 Estado del arte

Con el incremento de la importancia de los planogramas, nace la necesidad de verificar su cumplimiento. En un primer momento, la solución pasaba por la inspección por parte de una persona encargada de esta tarea, que apuntaba y realizaba informes en papel sobre el cumplimiento en cada uno de los puntos de venta. En los últimos años se ha sustituido el papel de los informes por dispositivos como teléfonos móviles o tabletas, con los que se puede añadir información y tomar fotografías. Aún así la tarea continúa siendo pesada y lenta de realizar, además de los errores que se pueden cometer al introducir los datos manualmente o al contar los productos a simple vista y de forma rápida para no perder un tiempo excesivo en la labor.

Las empresas que necesitan verificar el cumplimiento de los planogramas de sus productos optan por asignar la tarea de auditoría a sus propios delegados que visitan los diferentes comercios o contratar personas que se encarguen de hacer la gestión del punto de venta. Además, han surgido empresas que se encargan de ofrecer su propia red de GPVs en torno a un cierto territorio.

En la actualidad, tanto Pervasive Technologies (empresa en la que realizo prácticas) y Planorama se están desmarcando en el servicio de auditoría de planogramas ofreciendo una automatización para la detección de los productos en los lineales y expositores usando técnicas de Computer Vision junto a un análisis de los resultados de los planogramas. De tal modo se consigue una amplia recopilación de datos y una verificación de planogramas mucho más rápida y efectiva.

## 1.4 Requerimientos y funcionalidades

Conociendo ya el problema a resolver y la solución que se propone, se establece que en el sistema se distinguen dos grupos de usuarios clave: las personas responsables del Trade Marketing de la marca y los gestores de punto de venta, cada grupo con un papel diferenciado dentro de la aplicación. Es importante determinar cómo se utilizará la aplicación y qué funcionalidades son necesarias pensando en los casos reales de uso y en los objetivos que se desean obtener.

Para obtener datos reales de la usabilidad que debe ofrecer la aplicación para ambos perfiles de usuarios, ha sido posible organizar una entrevista a Fernando Albacar Ena, actual Sales Area Manager de la multinacional farmacéutica Bayer, dentro de la cual también ha trabajado coordinando tareas de gestión de puntos de venta y de Trade Marketing. Gracias a su colaboración, se ha creado una lista completa de las diferentes acciones que se deben poder realizar desde la aplicación.

### 1.4.1 Requisitos funcionales para gestores de puntos de venta

En la entrevista se concreta que, para las personas encargadas de gestionar el punto de venta, la principal función que deben poder realizar es capturar las imágenes de los lineales y expositores y cargarlas en la aplicación para que sean procesadas, añadiendo información relevante sobre el punto de venta en el que se ha tomado la imagen, la fecha y la hora de la visita.

También, se comentan posibles funcionalidades adicionales, como por ejemplo que se proporcione al usuario GPV una ruta completa y optimizada de los puntos de venta que debe visitar. Para esta primera fase del proyecto el sistema se ceñirá a realizar un seguimiento del cumplimiento de los planogramas. Sin embargo, la propuesta de proporcionar la ruta optimizada queda pendiente para una posible segunda fase, dado que el tiempo de desarrollo que implica es importante y no se encuentra dentro del alcance de este trabajo.

### 1.4.2 Requisitos funcionales para responsables de Trade Marketing

Por otro lado, las personas encargadas del Trade Marketing dentro de la empresa dispondrán de un conjunto más amplio de funcionalidades, ya que son las responsables de controlar que los planogramas acordados se estén cumpliendo y gestionar todo lo que esto implica.

La principal acción que se realizará desde el departamento de Trade Marketing es analizar las visitas de los gestores de punto de venta de la marca a los diferentes establecimiento. En el momento en que un GPV cargue una imagen en la aplicación, se podrán obtener inmediatamente los resultados del análisis del realograma y los datos de la visita. Más específicamente, para cada visita se podrá visualizar:

- ❖ Qué punto de venta se ha visitado.
- ❖ Quién ha realizado la visita.

- ❖ Cuando se ha hecho la visita. Fecha y hora.
- ❖ Imagen del lineal o expositor en el momento de la visita.
- ❖ Comparación entre la disposición real de los productos y el planograma contratado. Es decir, contrastar el planograma con el realograma.

Para realizar la evaluación y comparación del planograma contra el realograma, se generarán dos medidas que permitirán evaluar la calidad de la colocación de los productos en el punto de venta analizado. Estas dos métricas son el porcentaje de cumplimiento de la variedad de productos expuestos y el porcentaje de cumplimiento del área ocupada.

❑ Porcentaje de cumplimiento de la variedad de productos:

Muchas de las empresas que distribuyen sus productos a comercios ofrecen diferentes variedades de productos de la marca a los consumidores, diferentes SKUs. Por ejemplo, y para el caso de este proyecto, la marca de café L'OR distribuye a los supermercados las variedades de café RISTRETTO y ONYX, entre otros. Estos se exponen en distintos packagings<sup>1</sup>, diferentes cantidades de cápsulas, y con distintos precios. Para una marca, entonces, es muy importante conocer que las variedades de productos que se están exponiendo realmente en las estanterías son aquellas que se han acordado en el planograma.

Con la finalidad de obtener un indicador que sirva de evaluación del cumplimiento del planograma en función de la variedad de productos que se exhiben, se calcula un porcentaje. Este porcentaje es denominado "Porcentaje de cumplimiento de la variedad de productos" o PCVP. Para obtenerlo, se calcula el cumplimiento de cada una de las variedades de productos, dividiendo la cantidad que hay expuesta entre la cantidad que se ha contratado para ese SKU. Seguidamente se calcula la media de los cumplimientos de cada SKU y se transforma a porcentaje. Pongamos por ejemplo que en una visita de un GPV a un supermercado, se obtienen los resultados de la Tabla 1.1.

---

<sup>1</sup> Envoltorios, etiquetas y envases de los artículos comerciales que implican la inclusión o protección de productos para la distribución, almacenaje y venta (Sonneveld, 2000).

| Variedad de producto      | Cantidad real | Cantidad acordada |
|---------------------------|---------------|-------------------|
| RISTRETTO,<br>10 cápsulas | 10            | 20                |
| ONYX,<br>10 cápsulas      | 20            | 20                |
| COLOMBIA,<br>10 cápsulas  | 7             | 10                |
| DELIZIOSO,<br>10 cápsulas | 2             | 10                |

**Tabla 1.1:** Ejemplo de los resultados obtenidos tras la visita de un GPV a un punto de venta.

Entonces, el cumplimiento de cada producto se muestra en la Tabla 1.2:

| Variedad de producto      | Cumplimiento individual |
|---------------------------|-------------------------|
| RISTRETTO,<br>10 cápsulas | 0,5                     |
| ONYX,<br>10 cápsulas      | 1                       |
| COLOMBIA,<br>10 cápsulas  | 0,7                     |
| DELIZIOSO,<br>10 cápsulas | 0,2                     |

**Tabla 1.2:** Cumplimiento individual de cada variedad de productos, calculados a partir de la Tabla 1.1.

Tras hacer la media de los cumplimientos individuales y pasar el resultado a tanto por ciento, se obtiene un PVPC del 60%.

□ Porcentaje de cumplimiento de área:

Corresponde a la relación entre el área ocupada realmente por los productos de la marca en un expositor y el área contratada para ese expositor. Para calcularlo se divide el área real de los facings expuestos entre el área que ocupan los facings que se han acordado. Es importante, entonces, detectar las diferentes posiciones que puede adoptar un producto de la marca en los expositores, ya que si el producto exhibe en una caja rectangular, este mismo producto puede mostrar 6 facings distintos, correspondientes a cada una de las caras de la caja y que tienen superficies distintas.

En relación con posibles funcionalidades adicionales que podría tener la aplicación, se comentan las siguientes posibilidades:

- ❖ Lista de los puntos de venta en los que se comercializan productos de la marca.
- ❖ Información de cada uno de los puntos de venta. Por ejemplo:
  - Dirección
  - Teléfono de contacto
  - Correo electrónico de la persona responsable de cada punto de venta
  - Cómo ha evolucionado el cumplimiento de los planogramas en el punto de venta durante las últimas visitas.
- ❖ Posibilidad de contactar con el establecimiento por mail, dentro de la aplicación.
- ❖ Posibilidad de hacer una gestión de los usuarios que usan la aplicación, a los que se les debe poder aplicar roles, responsable de Trade Marketing y GPV.

## 1.5 Tecnología empleada

Una vez definidos el objetivo y alcance del proyecto que se desarrolla en este trabajo de final de grado, se pasa a analizar qué tecnologías hay disponibles a nuestro alcance para aplicarlas y conseguir la mejor solución posible para el sistema.

Se diferencian dos grandes grupos que conformarán el conjunto del sistema. Por un lado están las herramientas que componen el Pipeline Predictivo, el cual tiene la principal función de controlar y realizar el procesamiento de las imágenes que capturan los gestores de punto de venta, y por otro lado están las herramientas que componen la Aplicación Web, desde donde los usuarios podrán realizar la gestión y visualización de los resultados del procesado, junto a las demás funcionalidades. Ambas partes consumen diferentes servicios de Google Cloud Platform.

### 1.5.1 Google Cloud Platform

Google Cloud Platform, producto ofrecido por la compañía Google, se trata de una plataforma desde la que se ofrecen un conjunto de servicios de computación en la nube (cloud computing<sup>2</sup>) que se ejecutan en la misma infraestructura que usa la compañía para sus propios productos (como GoogleSearch, Gmail o Youtube) y cubren un gran número de casos de uso. Algunos servicios principales, como Compute Engine y Cloud Storage, permiten a los desarrolladores construir cualquier solución en un entorno virtualizado, evitando así la necesidad de desplegar infraestructura. La arquitectura de sistemas que utilizan este tipo de servicios en entornos virtualizados es conocida como “serverless”.

---

<sup>2</sup> Término asignado al uso de servidores remotos en internet para almacenar, administrar y procesar datos, y así evitar hacerlo en un servidor local o en el mismo ordenador personal.

Sin embargo, no es la única compañía que ofrece una plataforma de Cloud Computing de este tipo, ya que también existen grandes compañías como Amazon o Microsoft que presentan los servicios de las plataformas Amazon Web Services y Microsoft Azure, respectivamente. Aún así, para este proyecto se apuesta por las funciones disponibles en Google Cloud Platform debido a que es la única con la que he tratado anteriormente y los resultados y servicios obtenidos han sido muy buenos, así como también la documentación ofrecida. Además, al iniciarte en esta plataforma, se te proporcionan 300 dólares para gastar en ella y se utilizan durante el desarrollo de este proyecto.

En esta plataforma proveedora de nube pública se ofrece una lista de servicios muy amplia. A continuación se listan los 6 servicios de GCP que se utilizan en este proyecto, cada uno con funciones diferentes, las cuales se detallarán a medida que se avanza en la lectura:

- ❖ Google App Engine
- ❖ Google Cloud APIs
- ❖ Google Cloud Storage
- ❖ Google Cloud Firestore
- ❖ Google Cloud Functions
- ❖ Google Cloud AutoML Vision Object Detection

Además, en Google Cloud Platform todos los servicios que se ofrecen disponen de una interfaz gráfica desde la cual es posible mantener y administrar los diferentes dominios, cosa que hace más sencillo el uso de la plataforma.

Es importante destacar que para este proyecto también se tienen en cuenta los recursos económicos para garantizar la viabilidad. Por tanto, se intentan contratar servicios de Google Cloud Platform que se ajusten a las necesidades, pero sin excederse en recursos innecesarios.

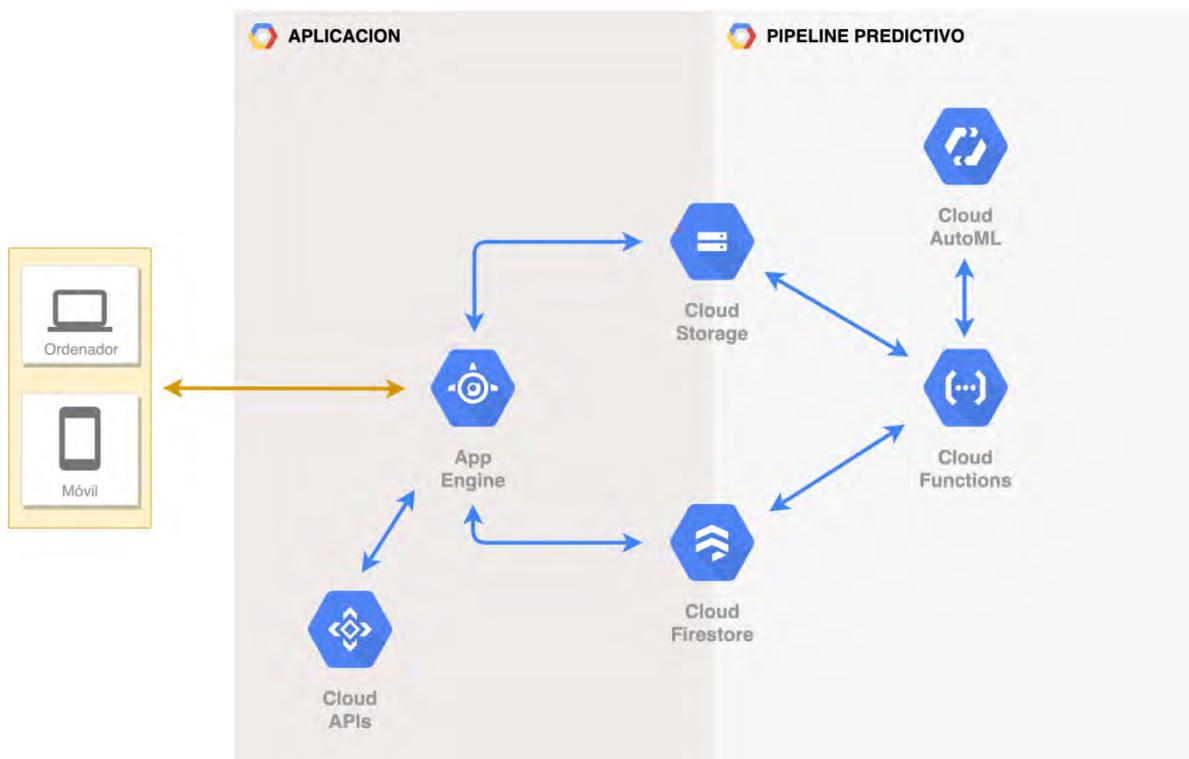
### 1.5.2 Diagrama de la arquitectura completa del sistema

El conjunto de la solución propuesta está formada por la agrupación de los servicios de Google Cloud Platform más los dispositivos desde los cuales se podrá acceder a la aplicación. En la Figura 1.4, se detalla visualmente la arquitectura completa del sistema para poder entender qué relaciones hay entre los distintos servicios de GCP que se utilizan y los dispositivos finales.

En primer lugar, desde los navegadores de dispositivos finales como ordenadores o teléfonos móviles se realizan las conexiones a AppEngine, núcleo del grupo “Aplicación”, y donde está almacenada la aplicación web y su servidor, encargado de ofrecer el código que el navegador ejecutará para mostrar la interfaz en la pantalla. App Engine, se conecta a las APIs de Google, Firestore y Google Storage. Por un lado, hace uso de alguna API como Google Maps y Geocoding para poder mostrar en la interfaz un mapa indicando donde se encuentran los puntos de venta. Por otro lado, AppEngine hace uso de Firestore para recibir, introducir y modificar datos de los registros o los

puntos de venta. Finalmente, también hace uso de Google Storage para almacenar y leer archivos disponibles por la aplicación, como son las imágenes que envían los GPVs o los resultados de la predicción que se han obtenido.

Por otro lado, el núcleo del grupo nombrado “Pipeline Predictivo” es el servicio de Cloud Functions. El sistema dispone de una única Cloud Function que está conectada a Cloud Storage, Cloud Firestore, al modelo de inteligencia artificial de Cloud AutoML Vision Object Detection. Por un lado, cuando un GPV publica una imagen de un expositor desde la aplicación web, esta es cargada al almacenamiento de archivos Cloud Storage. Es entonces cuando la Cloud Function, que está pendiente de los eventos de Storage, empieza a ejecutarse. Principalmente su tarea es leer la imagen que se ha introducido para seguidamente enviarla a AutoML Vision Object Detection, recoger los resultados de la predicción, calcular los porcentajes de cumplimiento del expositor de la imagen y guardar los datos en Firestore.



*Figura 1.4: Diagrama de la arquitectura completa del sistema.*

En los capítulos que siguen, se expone información detallada del sistema completo, empezando por el “Pipeline predictivo”, siguiendo con la explicación de la creación del modelo de inteligencia artificial usando Cloud AutoML Vision Object Detection, y detallando finalmente los servicios, tecnologías y pantallas de las que goza la aplicación final.



## 2. PIPELINE PREDICTIVO

En este proyecto se le asigna el nombre de Pipeline Predictivo al conjunto de acciones y procesos que se dan desde que el sistema recibe una imagen de un expositor hasta que se crea un archivo JSON con los resultados de la predicción, guardando también en la base de datos ambos porcentajes de cumplimiento. En esta sección, el objetivo es explicar detalladamente el funcionamiento de los servicios de Google Cloud Platform que conforman el Pipeline Predictivo y como son aprovechados en esta solución.

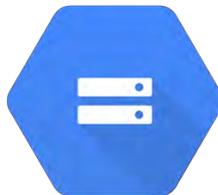
La lista de los servicios que se utilizan en el Pipeline Predictivo es la siguiente:

- ❖ Google Cloud Storage
- ❖ Google Cloud Firestore
- ❖ Google Cloud Functions
- ❖ Google Cloud AutoML Vision Object Detection

### 2.1 Google Cloud Storage

#### 2.1.1 Google Cloud Storage

Google Cloud Storage es un servicio de almacenamiento de archivos en línea para almacenar y acceder a datos desde la infraestructura de Google Cloud Platform. Este servicio combina el rendimiento y la escalabilidad de la nube de Google con capacidades avanzadas de seguridad y uso compartido.



*Figura 2.1: Icono de Google Cloud Storage.*

Los principales beneficios de Google Cloud Storage son:

- ❖ Almacenamiento de archivos en la nube de Google, con una accesibilidad garantizada del 99,99% al mes.
- ❖ Fácil gestión del control de acceso a los archivos. Estos pueden ser privados, públicos o compartidos.
- ❖ Los archivos son accesibles vía Google APIs o otras librerías de terceros.

Google Cloud Storage está organizado en Buckets (cubetas, traducido literalmente al castellano) identificados globalmente y únicos por el nombre al que se le asigna en un principio. Los Buckets contienen objetos o archivos a los cuales es posible acceder

mediante peticiones, siempre y cuando se disponga de una clave autorizada para hacerlo.

Además, el dominio de Google Cloud Storage ofrece diferentes soluciones de almacenamiento para adaptarse a la carga de trabajo que se quiera cubrir. En caso de necesitar un almacenamiento de archivos a los cuales se accede con muy alta frecuencia, se ofrecen estas dos posibilidades (Cloud, Google):

- ❖ Multi-Regional: discos duros repartidos por los diferentes datacenters de los que dispone Google repartidos por el mundo, optimizados para ofrecer la mejor redundancia geográfica<sup>3</sup> y una respuesta rápida a cualquier zona.
- ❖ Regional: discos almacenados en los datacenters de una región concreta seleccionable, también con redundancia dentro de la misma. Esta opción es más barata y se recomienda para aquellas soluciones que puedan permitirse un almacenamiento de datos en una región estrecha, lo que comporta un aumento ligero de la latencia cuando se accede a los datos desde regiones lejanas.

Por otro lado, para aquellos archivos a los que solamente se accede una vez al mes, o al año, se ofrecen respectivamente las opciones de Nearline o Coldline. Ofrecen un almacenamiento de archivos muy baratos y se suelen utilizar para guardar copias de seguridad de grandes proyectos o contenido multimedia poco frecuentado. Todo esto a cambio de una accesibilidad menor.

Para esta primera aproximación a ofrecer una solución en el cumplimiento de planogramas, utilizaremos la opción Regional de Storage, la cual ya nos será suficiente para los objetivos del proyecto.

## 2.1.2 Google Cloud Storage aplicado al proyecto

Aplicado al presentado en este trabajo, Google Cloud Storage se utiliza principalmente para almacenar:

- ❖ Imágenes que los gestores de punto de venta cargan al sistema mediante la aplicación web. Estas imágenes se guardan para poder ser procesadas y también visualizadas dentro de la interfaz.
- ❖ Resultados de la detección de productos en la imagen devueltos por el modelo de inteligencia artificial. Estos resultados se guardan en un archivo JSON que se deposita en Storage para así poder acceder a ellos a través de la aplicación y mostrar el cumplimiento del planograma.
- ❖ Archivos útiles para realizar el entrenamiento del modelo de inteligencia artificial. Estos archivos son un conjunto de imágenes y un archivo de tipo CSV, con las etiquetas de los productos en las imágenes. La relación entre ambos archivos se detalla en profundidad en el capítulo 3 del trabajo.

---

<sup>3</sup> Redundancia de los datos en diferentes espacios geográficos del planeta, con los objetivos de ofrecer respuestas rápidas a todo el mundo y evitar pérdidas y fallos en los datos.

## 2.2 Google Cloud Firestore

### 2.2.1 Cloud Firestore

Google Cloud Firestore es una base de datos basada en la nube con el sistema conocido NoSQL, construida para mantener la auto escalabilidad de los servidores que almacenan los datos y ofrecer muy buen rendimiento. Además, proporciona herramientas para conseguir una fácil integración de la base de datos en el desarrollo de aplicaciones, junto a una interfaz desde la cual se puede visualizar toda la información de la que se dispone.



*Figura 2.2: Icono de Google Cloud Storage.*

Las bases de datos NoSQL, a diferencia de las bases de datos SQL organizadas en tablas, están ordenadas en colecciones que contienen documentos rellenos con objetos JSON, donde está la información útil. El hecho de utilizar una base de datos NoSQL, conocida también como base de datos no relacional, permite que no haya ninguna restricción alrededor del tipo de datos se están introduciendo, ni errores cuando un mismo campo adquiere dos tipos de valores diferentes durante el transcurso de una operación (Xplenty, 2017). También, al disponer de un esquema flexible, los objetos dentro de una misma colección pueden tener campos diferentes, cosa que no se puede producir en las tablas de SQL.

Por otro lado, el hecho de ser una base de datos construida sobre la infraestructura de Google Cloud Platform, ofrece una escalabilidad automática de los recursos necesarios para cubrir cualquier necesidad, con la intención que solo se pague por aquello que realmente están usando. Más aún, los desarrolladores que usan Firestore no deben preocuparse de instalar ningún tipo de servidor ni configurar las conexiones con la red que esto conlleva. Tampoco de la seguridad de los servidores, ya que es la tecnología de Google la que se ocupa de cualquier posible ataque.

A su vez, al estar basada en la nube permite un acceso a los datos actualizados a tiempo real, manteniendo así todos los cambios disponibles para los usuarios.

Sin embargo, Firestore ofrece algunas limitaciones por lo que hace a las consultas de datos. Entre ellas, no es posible filtrar los datos por rango de diferentes campos, combinar consultas entre colecciones o utilizar el operador lógico OR o la cláusula “!=” en las comparaciones.

## 2.2.2 Google Cloud Firestore aplicado al proyecto

Para desarrollar la aplicación de planogramas se utiliza la tecnología de Firestore debido a, principalmente, la facilidad que nos ofrece para manejar los datos, además de otros beneficios. Por lo que hace a las limitaciones en las consultas de datos, en este proyecto no será necesario realizar consultas extremadamente complejas que justifique la implementación de una base de datos SQL.

Para ofrecer la solución descrita para este proyecto, son necesarias las colecciones “usuarios”, “establecimientos”, “registros” y “productos”. A continuación se detalla cada una de ellas:

### ❑ Usuarios

Colección en la cual se almacenan todos los datos del usuario que están registrados en el sistema y que permite realizar un control de acceso. Cada uno de los documentos de la colección corresponde a un usuario, y cada uno de los usuarios tiene los campos que se muestran en la Tabla 2.1:

| Campo       | Tipo de valor |
|-------------|---------------|
| acceso      | booleano      |
| displayName | string        |
| email       | string        |
| googleID    | string        |
| image       | string        |

**Tabla 2.1:** Formato de los documentos de la colección “usuarios”.

### ❑ Establecimientos:

Dispone de la lista de puntos de venta donde la marca tiene contratada la exhibición de sus productos. En esta colección de Firestore es posible recoger datos de la información general del punto de venta, como el nombre, la dirección, teléfono y correo electrónico de contacto. Además, cada punto de venta dispone de la distribución de los facings de los productos que se ha acordado, es decir, los facings de los que dispone el planograma negociado. Los facings son representados en un diccionario clave-valor, con el nombre del producto contratado y la cantidad del mismo que se debería exhibir.

Por otro lado, cuando un GPV realiza una visita a un punto de venta tomando una imagen del expositor/lineal y esta es procesada, los porcentajes calculados sobre el cumplimiento del planograma son añadidos a los datos del documento del mismo comercio visitado. Es decir, se añaden los campos de la Tabla 2.3 a la colección representada en la Tabla 2.2.

| Campo              | Tipo de valor |
|--------------------|---------------|
| nombre             | string        |
| provincia          | string        |
| poblacion          | string        |
| direccion          | string        |
| mapa               | geopoint      |
| telefono           | string        |
| correo_electronico | string        |
| facings_acordados  | map           |

**Tabla 2.2:** Formato de los documentos de la colección “establecimientos”.

| Campo                             | Tipo de valor |
|-----------------------------------|---------------|
| cumplimiento_porcentaje_area      | number        |
| cumplimiento_porcentaje_productos | number        |
| mejora                            | number        |

**Tabla 2.3:** Campos añadidos a la colección “establecimientos” tras procesar la imagen de la última visita realizada en ese establecimiento.

El campo mejora contiene un número y se calcula con la diferencia entre la media de los porcentajes de cumplimiento de la última visita y la media de los porcentajes de la visita procesada actual. Este valor se guarda para poder visualizar en la tabla de la interfaz si la visita que se ha realizado es mejor o no que la anterior.

#### ❑ Registros:

La colección “registros” contiene todas las visitas que los GPV han realizado a los diferentes comercios. El documento de cada visita contiene, entre otros campos, la identificación del punto de venta que se ha visitado, quién lo ha visitado, directorio a Google Storage de la imagen del expositor y los facings detectados en el planograma en el momento de la visita. Los facings acordados en el momento de la visita también se representan con un diccionario, igual que en la tabla de “establecimientos”.

Es importante guardar los facings acordados en el momento que se ha hecho cada una de las visitas ya que, en caso contrario, si el departamento de Trade Marketing decide cambiar los facings en el punto de venta, ya no habría forma de saber qué facings había acordados en las visitas anteriores al cambio.

El esquema completo de los documentos de esta colección está representado en la siguiente tabla, Tabla 2.4:

| Campo                       | Tipo de valor |
|-----------------------------|---------------|
| establecimiento_id          | string        |
| gpv_id                      | string        |
| fecha                       | fecha         |
| hora                        | string        |
| image_path                  | string        |
| facings_acordados           | map           |
| cantidad_facings_acordados  | number        |
| cantidad_facings_detectados | number        |

**Tabla 2.4:** Formato de los documentos de la colección “registros”.

#### ❑ Productos

En esta colección se almacena información de los productos que se pueden encontrar en los expositores de los comercios. Cada documento, que corresponde a un producto de la marca, contiene un único campo en el que especifican las dimensiones de la caja. Este dato es útil para calcular el porcentaje de cumplimiento del área contratada en el planograma.

| Campo  | Tipo de valor |
|--------|---------------|
| width  | number        |
| height | number        |
| depth  | number        |

**Tabla 2.5:** Formato de los documentos de la colección “productos”.

## 2.3 Google Cloud Functions

### 2.3.1 Google Cloud Functions

Google Cloud Functions es un servicio de computación en la nube del tipo Function as a Service (FaaS). Proporciona una plataforma con la que se puede desarrollar el código de funciones concretas que realiza procesos y conecta servicios, siempre sin la necesidad de construir ni mantener la arquitectura e infraestructura que sería necesaria en caso de no utilizar esta tecnología (Medium, 2018). Este hecho permite a los desarrolladores centrarse en el código.

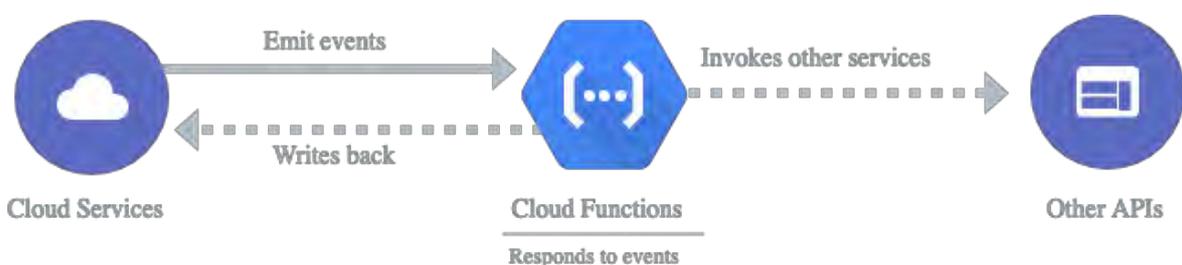


*Figura 2.3: Icono de Google Cloud Functions.*

Al empezar a utilizar Cloud Functions, es importante conocer los tres elementos principales en los que se basa:

- ❑ Eventos: suceso que ocurre dentro de Google Cloud Platform. Los tres eventos principales a los que una Cloud Function puede responder son:
  - Ficheros publicados, eliminados o modificados de Google Cloud Storage
  - Datos publicados, eliminados o modificados de Google Cloud Firestore
  - Invocación directa mediante una petición HTTP
- ❑ Trigger: Selección del evento que desencadenará la ejecución de la Cloud Function.
- ❑ Datos del evento: Conjunto de información que recibirá como parámetros de entrada la Cloud Function cuando se ejecute.

Entonces, a las Cloud Functions de Google se les asigna uno de los eventos listados que se producen dentro de la infraestructura de GCP. Cuando el evento asignado se desencadena, la Cloud Function emparejada responde, comenzando así su ejecución. Para ilustrar este funcionamiento, se adjunta la Figura 2.4, extraída de la documentación oficial de Google Cloud Platform (Google Cloud).



*Figura 2.4: Esquema de funcionamiento de las Cloud Functions de Google.*

En Google, las Cloud Functions pueden estar escritas en Node, Python o Go.

### 2.3.2 Google Cloud Functions aplicado al proyecto

Para el proyecto presente, el código que usa la Cloud Function de este sistema se desarrolla en Python. Esta dispone del trigger en Google Cloud Storage, de modo que cuando se suban archivos al Bucket de storage que se ha creado para el proyecto, se ejecutará esta función.

El proceso que sigue la Cloud Function desarrollada es:

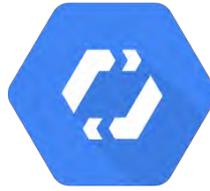
1. Confirmar que el archivo cargado a Google Storage es una imagen para continuar con la ejecución. En caso contrario se termina el proceso de la Cloud Function.
2. Procesar la imagen que se ha cargado a Google Storage para obtener los bytes en base64.
3. Enviar los bytes de la imagen al modelo de detección de objetos y así recibir la predicción de productos que se han detectado en la imagen.
4. Crear un archivo JSON con el resultado de la predicción, y guardarlo en Google Storage. El archivo JSON contiene las coordenadas de los recuadros que contienen un producto detectado dentro de la imagen, además de la clase asignada a cada bounding box.
5. Procesar el resultado de la predicción para calcular el cumplimiento de planogramas. En concreto, se calcula:
  - a. Porcentaje de cumplimiento de productos
  - b. Porcentaje de cumplimiento de área
  - c. Mejora de los porcentajes respecto el registro anterior en el punto de venta visitado
6. Publicar los tres cálculos anteriores en el documento correspondiente de Firestore.

## 2.4 Google Cloud AutoML Vision Object Detection

### 2.4.1 Google Cloud AutoML

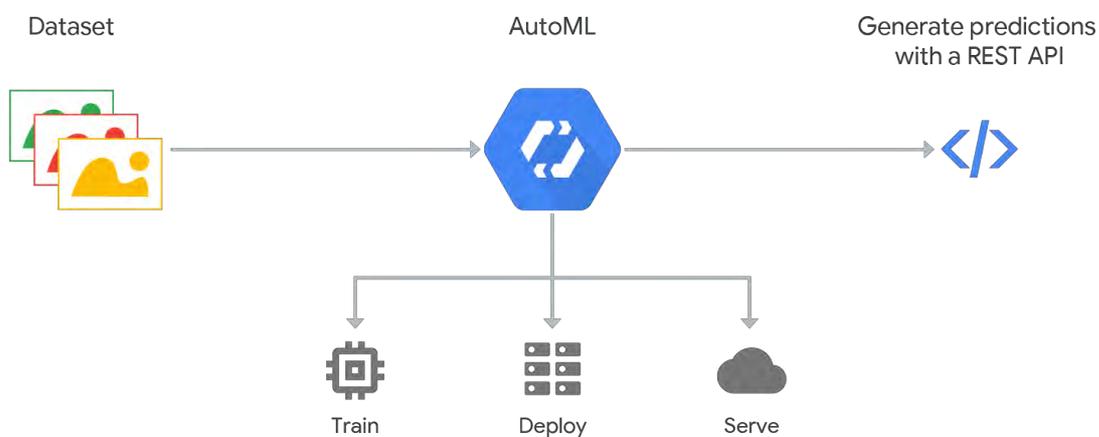
En la actualidad, la demanda de aplicaciones que usan Machine Learning (ML) crece continuamente, y no es de sorprender, ya que la cantidad de soluciones que puede llegar a ofrecer esta tecnología es inmensa. Sin embargo, la alta complejidad de tareas como el preprocesamiento de datos u optimización de los hiper parámetros del modelo que se está generando requieren de un tiempo elevado de desarrollo y supervisión. En este contexto, Google lanzó a su plataforma de servicios en la nube el producto conocido como Google Cloud AutoML, capaz de automatizar el diseño de la arquitectura de las redes neuronales para que los modelos resultantes se puedan aplicar fácilmente a distintas soluciones (Medium, 2018). Todo esto se desarrolla con la intención de que

todo el mundo pueda aplicar modelos de inteligencia artificial a sus proyectos e integrarlos a los demás servicios de Google Cloud Platform. Un ejemplo de caso de uso real de AutoML sería la clasificación de imágenes médicas en el proyecto presentado por Ekaba Bisong (Bisong, 2019).



**Figura 2.5:** *Icono de Google Cloud AutoML.*

En la Figura 2.6 se muestra un esquema general del funcionamiento de Google Cloud AutoML, extraído de la documentación oficial (Cloud, Google, 2019).



**Figura 2.6:** *Esquema de funcionamiento de AutoML.*

En cuanto a los servicios que ofrece, con AutoML se puede crear modelos especializados en imágenes, lenguaje natural y traducción. Para este proyecto se hace uso de AutoML Vision para sacar provecho de las imágenes que suben los gestores de punto de venta. En concreto, se hace uso del servicio Object Detection dentro de la sección Vision con el cual es posible crear un modelo de inteligencia artificial capaz de detectar objetos individuales dentro de las imágenes.

Además, la tecnología de AutoML está acompañada de una interfaz gráfica desde la cual es posible entrenar, evaluar y poner en funcionamiento los modelos de inteligencia artificial.

## 2.4.2 Google Cloud AutoML aplicado al proyecto

En cuanto a cómo se usa AutoML en este proyecto, y tal y como se ha ido comentando anteriormente, este se aplica para obtener una predicción que detecte los productos que están exhibidos en los lineales y expositores de las fotografías que cuelgan los GPVs. La detección de los productos hace posible el cálculo de los porcentajes de cumplimiento y así poder ofrecer una evaluación del cumplimiento del planograma.

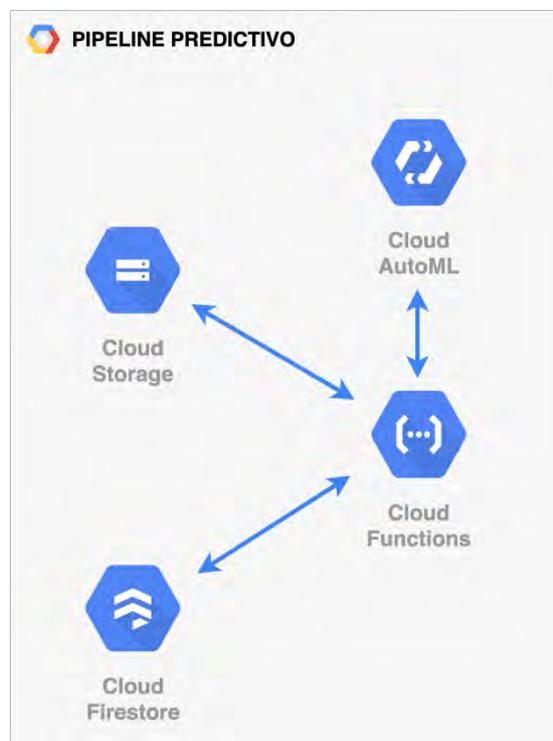
El proceso realizado para crear el modelo de inteligencia artificial mediante AutoML es muy importante para conseguir buenos resultados, es por eso que se detalla en profundidad en la sección 3.

## 2.5 Diagrama de la arquitectura del Pipeline Predictivo

Una vez que se conocen cada uno de los componentes que conforman el Pipeline Predictivo, es mucho más fácil entender la arquitectura que forman entre ellos. En la Figura 2.7 se puede observar esta arquitectura y las conexiones que hay entre los diferentes servicios.

Una síntesis del funcionamiento completo del Pipeline Predictivo es el siguiente:

1. El Pipeline Predictivo empieza su proceso una vez se ha colgado un archivo en Google Cloud Storage.
2. La Cloud Function, que su Trigger está pendiente de la subida de archivos a Storage, empieza a ejecutar el código Python. Las operaciones más importantes que realiza son:
  - a. Pasar la imagen al modelo de inteligencia artificial de AutoML.
  - b. Recibir, procesar y guardar los resultados de la predicción, tanto en Firestore como en Storage.



*Figura 2.7: Diagrama de la arquitectura del Pipeline Predictivo.*

### 3. DETECCIÓN DE OBJETOS

Una de las bases de la herramienta que se desarrolla en este proyecto es el sistema de reconocimiento y clasificación de objetos, basado en los modelos de inteligencia artificial autogenerados por el servicio de Google Cloud AutoML Vision. Sin un modelo de inteligencia artificial, sería muy difícil automatizar el proceso de la auditoría de los planogramas usando imágenes.

Tal y como se ha ido comentando, la detección estará centrada en productos de la marca de café L'OR. Se ha escogido esta marca debido a que los paquetes de los productos (el packaging que uno observa en los supermercados cuando va a comprar) son cajas de cartón firmes que raramente se encuentran deformadas cuando se exponen en las estanterías de los comercios. Para esta primera fase del proyecto es preferible un producto que normalmente mantenga el mismo aspecto en los lineales y expositores, antes que productos que sean fácilmente deformables como bolsas de patatas. Se hace esta elección puesto que se desconoce la exactitud que podemos llegar a obtener en la detección y clasificación y así evitar el coste de entrenamiento añadido que puedan causar los productos deformables.

No obstante, sí se ha podido observar en distintos puntos de venta que los productos de café L'OR pueden adoptar diferentes posiciones en lineal según han sido colocadas las cajas. Esto, que es necesario tenerlo en cuenta para crear las clases entre las que el sistema debe ser capaz de detectar, supone un reto. Más aún, esta marca también dispone de una larga variedad de productos que se distinguen principalmente por la intensidad de café o la cantidad de cápsulas que hay en la caja.

En este capítulo se detallan los procedimientos y recursos usados para elaboración de la herramienta que es capaz de realizar la detección y clasificación de productos L'OR en imágenes; desde la creación del conjunto de datos, hasta el entrenamiento y los resultados del modelo de inteligencia artificial. De este modo, en un futuro, se podrá cumplir con la idea y propósito de emular y extrapolar el mismo procedimiento que se aplica a los productos de la marca L'OR a cualquier otro tipo de producto de cualquier otra empresa.

#### 3.1 Creación del primer conjunto de datos

##### 3.1.1 Análisis de los productos de la marca L'OR

Como se ha comentado en la introducción del capítulo, la marca de café L'OR dispone de un largo surtido de distintas variedades de café. Para generar esta lista y analizar el aspecto de cada una de las cajas, primero nos fijamos en los productos que se ofrecen en la página oficial<sup>4</sup> de L'OR. En el momento de la visita a dicha página web, quitando los productos y ofertas solamente disponibles comprando online, las 26 variedades de café que habían disponibles están reflejados en la Tabla 3.1; en ella se muestran las variedades de café junto a los paquetes de cápsulas que ofrece la marca para cada una.

---

<sup>4</sup> Página oficial de L'OR: [https://www.lorespreso.com/es\\_es/](https://www.lorespreso.com/es_es/)

| Variedades de café      | Número de cápsulas en un paquete |
|-------------------------|----------------------------------|
| BARISTA SELECTION DOBLE | 10                               |
| COLOMBIA                | 10, 20                           |
| DECAFFEINATO            | 10                               |
| DELIZIOSO               | 10                               |
| DOBLE FORZA             | 10                               |
| DOBLE LUNGO PROFONDO    | 10                               |
| DOBLE RISTRETTO         | 10                               |
| DOBLE SPLENDEnte        | 10                               |
| FORTISSIMO              | 10                               |
| FORZA                   | 10                               |
| GRAN CAFÉ FILTRO        | 10                               |
| INDIA                   | 10                               |
| LUNGO ELEGANTE          | 10                               |
| LUNGO ESTREMO           | 10                               |
| LUNGO PROFONDO          | 10                               |
| MATTINATA LUNGO         | 10                               |
| ONYX                    | 10, 20, 40                       |
| OR ABSOLU               | 10                               |
| OR NOIR                 | 10                               |
| OR ROSE                 | 10                               |
| PAPÚA NUEVA GUINEA      | 10                               |
| RISTRETTO               | 10, 20, 40                       |
| RISTRETTO DECAFFEINATO  | 10, 20                           |
| SATINATO                | 10                               |
| SONTUOSO                | 10, 20                           |
| SPLENDEnte              | 10                               |
| SUPREMO                 | 10                               |

**Tabla 3.1:** SKUs de la marca L'OR disponibles en la página web.

Pueden haber paquetes de 10, 20 y hasta de 40 cápsulas para aquellas variedades de productos más vendidas, como es el caso de RISTRETTO u ONYX. Por tanto, 34 es la cantidad de paquetes distintos, teniendo en cuenta este hecho. En otras palabras, se contabilizan 34 SKUs.

Para poder visualizar cómo son y cual es la diferencia entre las diferentes tamaños de cajas según las cápsulas que contengan, en las Figuras 3.1, 3.2 y 3.3 se exponen imágenes de algunos de los paquetes que se pueden encontrar en los supermercados:



*Figura 3.1: Ejemplo de paquetes de café L'OR de 10 cápsulas. De izquierda a derecha: RISTRETTO, ONYX y SONTUOSO.*



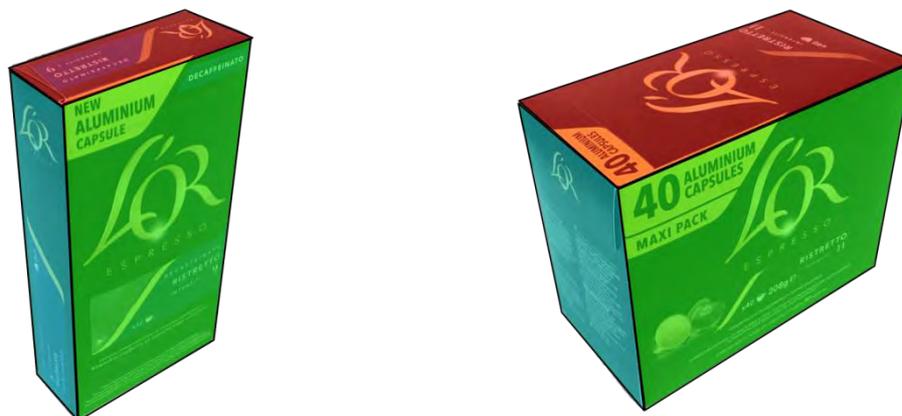
*Figura 3.2: Ejemplo de paquetes de café L'OR de 20 cápsulas. De izquierda a derecha: RISTRETTO, ONYX y SONTUOSO.*



*Figura 3.3: Ejemplo de paquetes de café L'OR de 40 cápsulas. De izquierda a derecha: RISTRETTO y ONYX.*

Así mismo, el sistema ha de ser capaz de detectar cajas en la imagen y clasificar cada una de ellas entre 34 SKUs diferentes. Pero no sólo eso, también se quiere detectar cuál de las 6 caras del cubo que forma el paquete es visible para el consumidor. La cara visible es significativa debido a la importancia que tiene para calcular el área ocupada por productos de la marca y compararla con el área contratada en el planograma. En función de la cara expuesta, el paquete ocupará frontalmente en el expositor más o menos área. Esto nos lleva a tener una cantidad de 102 facings distintos posibles y por

lo tanto también 102 clases entre las que el modelo debe ser capaz de clasificar un producto detectado. El número 102 resulta de multiplicar la cantidad de distintos SKUs (34) por las 3 caras distintas que puede tener el paquete, considerando que las caras opuestas son iguales<sup>5</sup>. Las tres caras distintas son la frontal, la lateral y la superior.



*Figura 3.4: Cara frontal (verde), lateral (azul) y superior (roja) en cajas L'OR de 10 y 20 cápsulas.*

### 3.1.2 Captura de imágenes

Al comenzar el proyecto se disponía de fotografías de lineales y expositores con productos de la marca L'OR, pero eran insuficientes para realizar un entrenamiento exitoso. Por esto, fueron necesarias las visitas a diferentes supermercados para captar más imágenes. Cuantos más supermercados diferentes se visitaran para captar imágenes, más expositores de distinta iluminación se obtendrían (muy conveniente para entrenar el sistema), por tal razón pedí colaboración a compañeros, amigos y familiares para que, en el momento de hacer la compra, tomaran imágenes de los productos de la marca en cuestión. Agrupando todas las imágenes se pudo conseguir un primer conjunto de datos con fotografías hechas desde diferentes teléfonos móviles y, por tanto, cámaras con resoluciones diferentes.

Los principales criterios que se intentan seguir en el momento de tomar la imagen es que los productos L'OR estén, en la manera de lo posible, en posición frontal al objetivo de la cámara y, además, que entren todos dentro de la imagen. No obstante, no siempre se es capaz de hacer fotografías de los lineales/expositores de la tal manera debido a los pasillos estrechos de algún punto de venta, a la cantidad de clientes que puede haber u obstáculos lo que dificultan. Por lo cual, para crear el conjunto de datos también se tomaron algunas imágenes con perspectiva y desde lejos para simular estas posibles situaciones que se pueden encontrar los GOVs durante una visita. Algunos ejemplos de imágenes captadas se presentan en las Figuras 3.5 y 3.6.

---

<sup>5</sup> Consideramos que las caras opuestas de la caja son iguales dado que ocuparan el mismo espacio y computarán como la misma área en el momento de hacer el cálculo de área ocupada por productos de la marca.



*Figura 3.5: Imágenes de expositores que se han podido realizar frontalmente.*



*Figura 3.6: Imágenes de expositores que no se han podido realizar frontalmente.*

Durante esta primera acumulación de fotografías se consiguieron, tras eliminar imágenes repetidas o muy parecidas, un total de 62 imágenes útiles.

### 3.1.3 Etiquetado

Normalmente, un sistema de reconocimiento automático de imágenes implica un etiquetado previo de las imágenes que se usan para entrenar el sistema. El etiquetado manual de imágenes es el proceso de definir manualmente regiones de una imagen y asignar una descripción (etiqueta) a cada región. Según los objetivos que se buscan y los datos que se quieren conseguir, existen diferentes posibilidades para realizar el etiquetado. Por ejemplo, se puede etiquetar una imagen asignándole a la imagen entera una etiqueta concreta o segmentar la imagen en partes y añadir etiquetas a cada una de estas partes. La segmentación se puede hacer con rectángulos (*bounding boxes*<sup>6</sup>), polígonos, puntos, líneas o píxel a píxel.

<sup>6</sup> En procesamiento digital de imágenes, las *bounding boxes* son coordenadas de un polígono rectangular situado en una zona de la imagen que, normalmente, envuelve objetos que aparecen en ella.

Para este proyecto en concreto necesitamos etiquetar la imagen segmentándola en *bounding boxes*, enmarcando cada uno de los productos de café L'OR que aparecen en las imágenes y asignándoles una etiqueta. Los datos generados, junto a las imágenes, será aquello que se utilizará para entrenar el modelo de aprendizaje automático de AutoML Vision Object Detection. En lo que sigue se expone el proceso y herramientas usadas para elaborar todo el proceso.

### 3.1.3.1 Herramienta utilizada

En la actualidad, existen varias plataformas que ofrecen múltiples funcionalidades complementarias para realizar el etiquetado. Y es que cualquier funcionalidad, por poco que ayude puede acabar comportando una gran reducción del tiempo de etiquetado cuando se trata de muchas imágenes.

Para este proyecto, basándose en los criterios de precio, herramientas, formatos y facilidad de uso, se escoge el software LabelImg (Loop, 2018). Como describen en su propia página pública de Github<sup>7</sup>, LabelImg es una herramienta gráfica de etiquetado de imágenes usando recuadros. Es una herramienta sin coste, de código abierto, escrita en Python, con la única posibilidad de etiquetar usando *bounding boxes*.

Una vez LabelImg ya está instalado, es importante configurar cuales son las clases (o etiquetas) entre las cuales es posible catalogar las *bounding boxes* que se crearán. Para ello lo que debemos hacer es modificar el archivo “predefined\_classes.txt”, que se encuentra en la carpeta “data”, situada dentro del directorio donde se ha instalado LabelImg. El proceso es tan simple como introducir en el archivo de texto mencionado la lista de posibles clases entre las cuales se puede clasificar un objeto encontrado en las imágenes, cada una en una nueva línea. Una buena práctica es escribir el nombre de las clases sin espacios. De tal forma y tomado como ejemplo, para el paquete de 10 cápsulas del producto RISTRETTO DECAFFEINATO se escribirán las líneas:

- ❖ RISTRETTO\_DECAFFEINATO\_10
- ❖ RISTRETTO\_DECAFFEINATO\_10/2
- ❖ RISTRETTO\_DECAFFEINATO\_10/4

El número 10 (en este caso, pero que también puede ser 20 o 40) indica la cantidad de cápsulas del paquete. Por otro lado, las clases que finalizan en “/2” indican que la parte visible en la imagen es la cara lateral del producto, y los que finalizan en “/4” la cara superior. Esta nomenclatura es la nueva forma que tendremos de identificar todos los distintos facings de la marca L'OR que se pueden encontrar en un punto de venta.

El siguiente paso ya es empezar a etiquetar cada una de las imágenes, envolviendo cada producto de la marca L'OR en un rectángulo e indicando a cuál de las clases pertenece. Para ilustrar mejor este proceso, en la Figura 3.7 se muestra un ejemplo de etiquetado de de una de las imágenes recogidas usando la interfaz de LabelImg.

---

<sup>7</sup> Página de Github de LabelImg: <https://github.com/tzutalin/labelImg>



*Figura 3.7: Interfaz del programa Labellng con el etiquetado realizado sobre la imagen, que solamente tiene dos productos de café L'OR.*

```

1 <annotation>
2 <folder>planogramas_20190426</folder>
3 <filename>IMG_3468.jpg</filename>
4 <path>/Users/polaibacar/Desktop/TFG/imagenes planogramas/planogramas_20190426/IMG_3468.jpg</path>
5 <source>
6 <database>Unknown</database>
7 </source>
8 <size>
9 <width>4032</width>
10 <height>3024</height>
11 <depth>3</depth>
12 </size>
13 <segmented>0</segmented>
14 <object>
15 <name>RISTRETTO_DECAFFEINATO_10</name>
16 <pose>Unspecified</pose>
17 <truncated>0</truncated>
18 <difficult>0</difficult>
19 <bndbox>
20 <xmin>3012</xmin>
21 <ymin>1188</ymin>
22 <xmax>3681</xmax>
23 <ymax>2473</ymax>
24 </bndbox>
25 </object>
26 <object>
27 <name>SONTUOSO_10</name>
28 <pose>Unspecified</pose>
29 <truncated>0</truncated>
30 <difficult>0</difficult>
31 <bndbox>
32 <xmin>419</xmin>
33 <ymin>1186</ymin>
34 <xmax>1090</xmax>
35 <ymax>2471</ymax>
36 </bndbox>
37 </object>
38 </annotation>
39

```

*Figura 3.8: Ejemplo de archivo XML que retorna el software Labellng al etiquetar una imagen. Este ejemplo corresponde a la imagen etiquetada mostrada en la Figura 3.7.*

Por defecto, el formato de los datos que genera este software es PascalVOC XML, aunque también se puede configurar para generar el formato YOLO. PascalVOC XML es un formato estándar de representar las posiciones en las que se encuentran los diferentes objetos en la imagen. De esta forma, al etiquetar haciendo recuadros dentro de la imagen se genera, para cada instantánea, un archivo XML con el formato que sigue el ejemplo de la Figura 3.8 (surgido del etiquetado de la Figura 3.7).

El archivo dispone del elemento principal “anotation”; dentro de este se encuentra las diferentes informaciones que tenemos de la imagen. Primeramente se detallan las características principales del archivo que se ha etiquetado, como son la carpeta en la que se encuentra la imagen (elemento “folder”), el nombre del archivo (“filename”) y el directorio completo de la imagen (“path”). A continuación se indica el tamaño de la imagen, anchura (“width”) y altura (“height”) dentro del elemento “size”.

En último lugar se encuentran los elementos “object”, uno por cada recuadro que se ha dibujado en la imagen del expositor. Dentro de este se encuentran más elementos con la información de la etiqueta. En la lista siguiente se muestran los elementos que serán útiles en el momento de procesar los archivos XML:

- ❖ Elemento “name”: texto que indica el nombre de la etiqueta que se le ha asignado a ese recuadro.
- ❖ Elemento “truncated”: booleano (0 o 1) que indica cuando un objeto de la imagen no es completamente visible en esta debido a que no ha entrado completamente en el perímetro de la imagen y, pese a esto, se enmarca con un recuadro igualmente. Este valor se establece automáticamente
- ❖ Elemento “difficult”: booleano (0 o 1) que sirve para indicar que el objeto etiquetado es difícilmente visible en la imagen y puede que cause confusiones al modelo. Esta opción es seleccionable desde la interfaz.
- ❖ Elemento “bndbox”: recoge la posición de la *bounding box* en la imagen. La posición se indica con los elementos “xmin”, “ymin”, “xmax” y “ymax” que se encuentran dentro de este mismo elemento.
  - Elemento “xmin”: indica la coordenada horizontal de la esquina superior izquierda del recuadro.
  - Elemento “ymin”: indica la coordenada vertical de la esquina superior izquierda del recuadro.
  - Elemento “xmax”: indica la coordenada horizontal de la esquina inferior derecha del recuadro.
  - Elemento “ymax: indica la coordenada vertical de la esquina inferior derecha del recuadro.

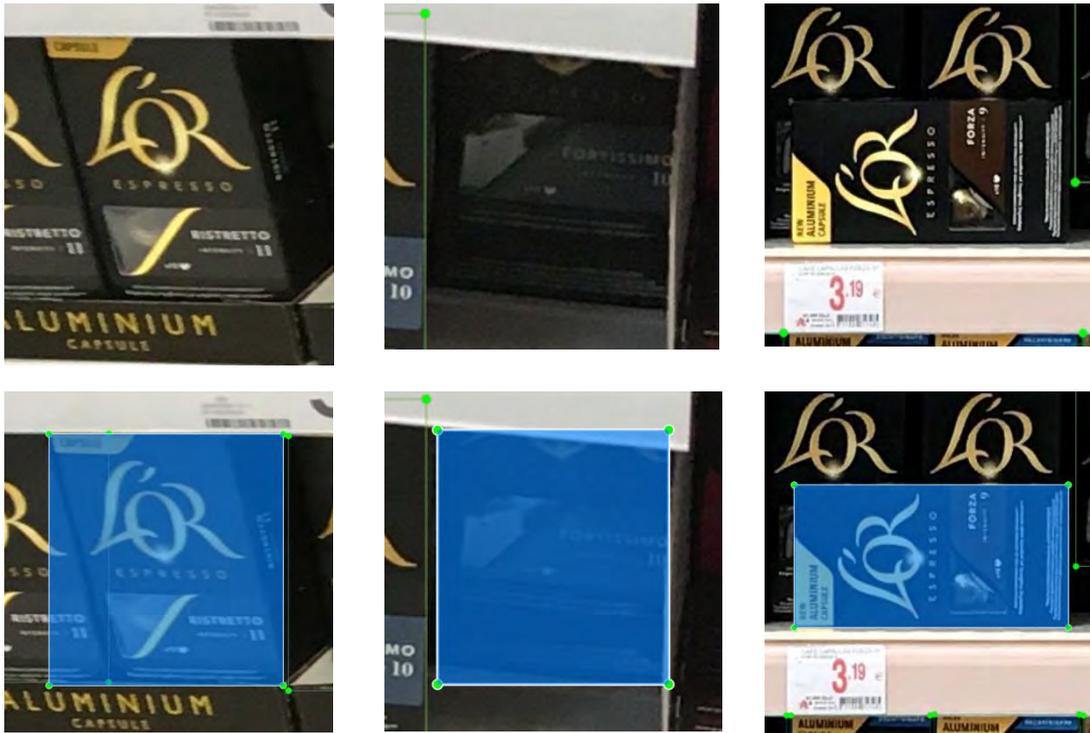
Las cuatro coordenadas que se encuentran en el elemento “bndbox” son mínimas necesarias para poder conocer dónde se sitúa el recuadro dentro de la imagen.

Es importante destacar el hecho de que poder indicar cuando una etiqueta es de un producto que no ha cabido en la imagen (“truncated”), o que el producto es difícilmente visible (“difficult”) nos es útil para tener controlados estas bounding boxes que, en el momento del entrenamiento, pueden causar problemas. De tal forma, en caso de hacer un primer entrenamiento y que este no obtenga muy buenos resultados, cabría la posibilidad de eliminar las etiquetas que estén marcadas como “truncated” y/o “difficult” y probar de nuevo a entrenar el modelo.

### 3.1.3.2 Criterios a seguir en el etiquetado

Con la intención de conseguir el mejor *dataset* posible, se fijan los siguientes criterios de etiquetado:

- ❖ El rectángulo que se dibuja debe ajustarse todo lo posible al contorno del producto, no se debe dejar margen innecesario. Hay que tener en cuenta que, dependiendo de la perspectiva de la imagen, no siempre es posible cuadrar el rectángulo con el producto a la perfección. En ese caso, siempre se intenta que entre toda la cara del producto en la etiqueta.
- ❖ Hay situaciones en las una caja de café L’OR se ve con dificultad, ya sea porque está al fondo de un expositor y no le da la luz o porque está parcialmente tapado por otro producto o por una etiqueta de ofertas. En estos casos, es decisión de la persona que etiqueta la imagen decidir si puede causar confusión al modelo y, en ese caso marcar la casilla “difficult” del LabelImg para la etiqueta en concreto. Esta casilla está situada en la parte superior derecha de la interfaz.
- ❖ Los productos que están cortados por el tamaño de la imagen también se etiquetan.
- ❖ Los productos que se encuentran tumbados o boca abajo se etiquetan igual que cualquier otro producto puesto de forma natural.



**Figura 3.9:** Ejemplos de etiquetado de imágenes con productos L'OR en situaciones dudosas.

### 3.1.3.3 Resultados del etiquetado

Al finalizar el proceso de etiquetado, se pudo notar que, en las fotografías que se tomaron, no aparecían muchos de los productos L'OR que se habían listado en un principio. Las clases que se encontraron en las imágenes fueron 34 de las 102 posibles. Esto quiere decir que entrenar el modelo con estos datos supondrá que solamente sea capaz de detectar y clasificar 34 clases.

### 3.1.4 Subir las imágenes a Google Cloud Storage

Una vez ya se dispone de todos los archivos de tipo XML creados por el programa etiquetador de todas las imágenes acumuladas, el siguiente paso es subir estas imágenes a Google Cloud Storage.

Es importante saber que los tipos de archivos soportados para el entrenamiento de AutoML son JPEG, PNG y GIF y el tamaño máximo de las imágenes es de 30MB. También es importante saber que AutoML, para realizar el entrenamiento, redimensiona las imágenes que superan los 1024 por 1024 píxeles y las ajustan a este tamaño. Por tanto, las imágenes que superen estas dimensiones pueden sufrir una pérdida de calidad. Esto puede afectar a algunas *bounding boxes*, ya que también se debe cumplir que ninguna de ellas sea más pequeña que 8 por 8 píxeles.

### 3.1.5 Transformación de los datos para adaptarlos a AutoML

Para continuar con el proceso y acabar entrenando el modelo, es necesario entender cómo se le indican a la plataforma de AutoML las coordenadas de las *bounding boxes* y qué etiqueta tiene cada una, es decir, qué clase se les ha asignado. Como se ha visto, los archivos XML generados en el proceso de etiquetado disponen de esta información. Sin embargo, AutoML requiere que se introduzcan estos datos siguiendo un formato determinado. En concreto, pide que se suba un único archivo de tipo CSV donde esté toda la información. Cada fila del archivo ha de representar una *bounding box*, y todas deben seguir el formato de los siguientes ejemplos, cogidos de la propia documentación de Google:

```
TRAIN,gs://folder/image1.png,car,0.1,0.1,,,0.3,0.3,,
TRAIN,gs://folder/image1.png,bike,.7,.6,,,,.8,.9,,
UNASSIGNED,gs://folder/im2.png,car,0.1,0.1,0.2,0.1,0.2,0.3,0.1,0.3
TEST,gs://folder/im3.png,,,,,,,,,
```

Es importante mantener las comas como separador de columnas. El significado de cada columna se detalla ordenadamente a continuación:

1. A qué grupo de datos se asigna la *bounding box*: “TRAIN”, “VALIDATION” o “TEST”. Cuando se entrena un modelo de “Machine Learning”, comúnmente se separan los datos en los tres grupos mencionados.

El grupo de “TRAIN” se utiliza para construir el modelo. Cuando empieza el entrenamiento, el modelo prueba diferentes parámetros mientras busca patrones en los datos de “TRAIN”. En el momento en que el modelo identifica patrones, usa los datos del grupo “VALIDATION” para evaluar los parámetros que se habían ajustado. Seguidamente, cuando los patrones y parámetros con mejores resultados ya han sido identificados, se usa el grupo de “TEST” para evaluar el modelo creado calculando la tasa de error, la calidad y la exactitud que este puede llegar a proporcionar.

También hay la posibilidad de utilizar el valor “UNASSIGNED” en esta primera columna en lugar de “TRAIN”, “VALIDATION” o “TEST”. De tal modo, AutoML Vision Object Detection asignará aleatoriamente a cada *bounding box* uno de los tres grupos mencionados. La asignación no es completamente aleatoria, sino que el 80% de las *bounding boxes* se utilizarán en “TRAIN”, 10% en “VALIDATION” y el 10% restante en “TEST”. Más aún, hay que tener en cuenta que las *bounding boxes* de una misma imagen irán todas al mismo grupo. Para este proyecto en concreto se emplea la segunda opción, utilizar el valor “UNASSIGNED” para todas las filas del archivo CSV.

2. Directorio de Google Storage donde se encuentra la imagen en concreto a la que pertenece la *bounding box* de la fila.
3. Clase que se le ha asignado a la *bounding box*. En este caso serán los diferentes nombres de productos de la marca que se han asignado en el momento del etiquetado.

4. Coordenadas de la *bounding box* en la imagen, separadas por comas. La *bounding box* puede estar especificada de dos maneras.

- a. Indicando solamente dos vértices diagonalmente opuestos del recuadro. Éste será nuestro caso, ya que en el archivo XML disponemos de las coordenadas del vértice superior izquierdo y del vértice inferior derecho. El formato es el siguiente:

```
xmin,ymin,,xmax,ymax,,
```

- b. Indicando los 4 vértices del recuadro. Entonces, el formato sería:

```
xmin,ymin,xmax,ymin,xmax,ymax,xmin,ymax
```

Otro aspecto importante a tener en cuenta para indicar correctamente las coordenadas, es que estas deben estar indicadas con un valor entre 0 y 1, teniendo en cuenta que el punto (0,0) representa la esquina superior izquierda de la imagen y (1,1) representa la esquina inferior derecha.

Sabiendo esto ya se es capaz de empezar a construir el archivo CSV con el formato requerido. Como es evidente, construirlo manualmente requeriría de mucho esfuerzo, es por eso que se usa un código escrito en Python para aprovechar la organizada información de la que se dispone en los archivos XML generados por el LabelImg. El código Python mencionado tiene la principal función de automatizar el proceso de leer la información de todos los archivos XML generados en el etiquetado y crear un archivo CSV con la información recogida y el formato adecuado.

Además, en este código de Python se añade un contador para conocer la cantidad de bounding boxes que se han etiquetado en las 62 imágenes. La respuesta es 1303. Una media de 21 productos L'OR por imagen.

### 3.1.6 Importar el archivo CSV a AutoML

Una vez ya tenemos el archivo CSV que representa nuestros datos, es el momento de subirlo a Cloud Storage para, seguidamente, importarlo a AutoML. Para ello, mediante la interfaz gráfica que proporciona Google para AutoML Vision Object Detection, importamos el CSV indicando el directorio completo de Google Storage donde se encuentra el archivo. Este proceso puede llegar a tardar un rato, dependiendo de la cantidad de filas que tenga el CSV.

Además, es posible que aparezca algún error, indicando que alguna de las *bounding boxes* etiquetadas es más pequeña a 8 por 8 píxeles debido a la redimensión que aplica AutoML a las imágenes que superan los 1024 píxeles de ancho o largo.

### 3.1.7 Eliminar etiquetas escasas

Tras la importación, en la interfaz gráfica se muestra una tabla con la distribución que ha hecho AutoML de los datos entre los grupos “TRAIN”, “VALIDATION” y “TEST” de cada una de las clases que se han etiquetado en las imágenes. En la Tabla 3.2 se muestra esta distribución.

No obstante, en AutoML se establece un requisito mínimo de 10 *bounding boxes* por cada clase para poder entrenar el modelo y que los resultados sean mínimamente satisfactorios. En concreto deben haber, como mínimo, 8 ejemplos en “TRAIN”, 2 en “VALIDATION” y 2 en “TEST”. Aquellas filas marcadas en rojo en la Tabla 3.2 corresponden a las clases de las cuales no se dispone suficientes *bounding boxes* como para llegar al requisito mínimo. Estas deben ser eliminadas para poder continuar y realizar el entrenamiento pero hay que tener presente que el modelo no será capaz de detectarlas ni clasificarlas.

| CLASE                       | TRAIN | VALIDATION | TEST | TOTAL |
|-----------------------------|-------|------------|------|-------|
| COLOMBIA_20                 | 1     | 0          | 0    | 1     |
| ONYX_40                     | 1     | 0          | 0    | 1     |
| RISTRETTO_DECAFFEINATO_20/2 | 1     | 0          | 0    | 1     |
| FORZA_10/2                  | 0     | 1          | 1    | 2     |
| ONYX_10/2                   | 1     | 1          | 0    | 2     |
| PROFONDO_LUNGO_10/2         | 1     | 1          | 0    | 2     |
| RISTRETTO_10/4              | 2     | 0          | 0    | 2     |
| RISTRETTO_40                | 2     | 0          | 0    | 2     |
| SOUNTUOSO_20                | 2     | 0          | 0    | 2     |
| RISTRETTO_20                | 3     | 1          | 1    | 5     |
| SPLENDETE_10/2              | 1     | 2          | 2    | 5     |
| DELIZIOSO_10/4              | 5     | 2          | 1    | 8     |
| ESPRESSO_DECAFFEINATO_10/4  | 5     | 2          | 1    | 8     |
| ONYX_20/2                   | 7     | 2          | 1    | 10    |
| RISTRETTO_DECAFFEINATO_10/4 | 6     | 3          | 1    | 10    |
| RISTRETTO_10/2              | 13    | 0          | 0    | 13    |
| SPLENDETE_10/4              | 10    | 5          | 2    | 17    |
| SONTUOSO_10/4               | 9     | 6          | 3    | 18    |
| RISTRETTO_DECAFFEINATO_20   | 15    | 5          | 5    | 25    |
| FORZA_10/4                  | 16    | 6          | 4    | 26    |

|                           |    |    |    |     |
|---------------------------|----|----|----|-----|
| PROFONDO_LUNGO_10/4       | 23 | 4  | 4  | 31  |
| PROFONDO_LUNGO_10         | 20 | 18 | 6  | 52  |
| DELIZIOSO_10              | 46 | 9  | 8  | 63  |
| FORZA_10                  | 42 | 14 | 10 | 66  |
| ONYX_20                   | 39 | 21 | 8  | 68  |
| RISTRETTO_20/2            | 38 | 23 | 9  | 70  |
| ESPRESSO_DECAFFEINATO_10  | 50 | 15 | 8  | 73  |
| COLOMBIA_10               | 52 | 9  | 13 | 74  |
| RISTRETTO_DECAFFEINATO_10 | 57 | 18 | 15 | 90  |
| ONYX_10                   | 60 | 22 | 13 | 95  |
| SPLENDENTE_10             | 67 | 22 | 12 | 101 |
| SONTUOSO_10               | 60 | 24 | 18 | 102 |
| FORTISSIMO_10             | 67 | 24 | 14 | 105 |
| RISTRETTO_10              | 93 | 38 | 21 | 152 |

**Tabla 3.2:** Distribución entre “TRAIN”, “VALIDATION” y “TEST” de cada una de las clases. Listadas en orden ascendente según la cantidad de bounding boxes disponibles.

Tras eliminar las etiquetas de las clases que no cumplen con los requisitos mínimos, quedan disponibles para el entrenamiento 1229 bounding boxes y 18 clases distintas.

### 3.2 Primer entrenamiento

Tras obtener imágenes de expositores con productos de la marca L’OR, etiquetarlas y procesar los datos para adaptarlos a AutoML, el siguiente paso ya es entrenar el modelo.

Al presionar el botón “TRAIN” de la interfaz gráfica de AutoML Object Detection, se abre una ventana emergente en la que podemos escoger entre dos opciones:

1. Optimizar el modelo para conseguir una mayor exactitud en las predicciones, pero con mayor tiempo de respuesta, entre 800ms y 1500ms.
2. Optimizar el modelo para conseguir un tiempo de respuesta menor, entre 300ms y 500ms, pero con menor exactitud.

En el proyecto en el que se trabaja, para ofrecer el mejor servicio posible, requiere de precisión por delante de velocidad. Entonces, se escoge la primera opción. Seguidamente el modelo comienza la fase de entrenamiento.

Tras 2 horas y 44 minutos, el modelo acabó el entrenamiento. Supone un coste aproximado de \$85. El cálculo está hecho contando las horas de entrenamiento como 3,

ya que se contabilizan las horas comenzadas, multiplicadas por \$28.35 que cuesta cada hora de entrenamiento.

### 3.3 Evaluación del primer modelo

Una vez el modelo ya está listo, el primer paso es analizar la propia evaluación que hace AutoML usando las imágenes que han sido añadidas en el grupo “TEST”.

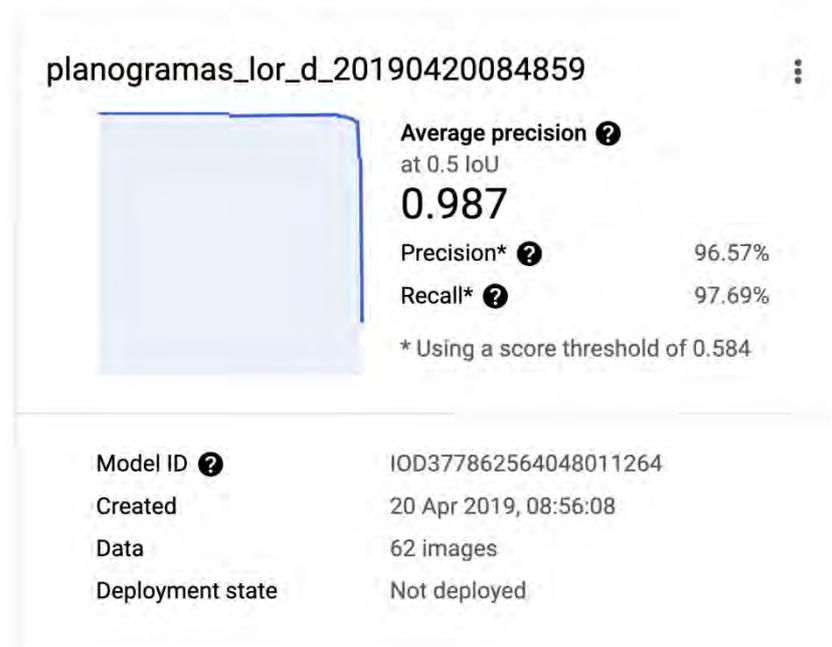


Figura 3.10: Evaluación de AutoML del primer modelo.

En la Figura 3.10, se puede observar dicha evaluación. El modelo ha conseguido una precisión<sup>8</sup> media del 98,7% calculada con un valor IoU del 50%. También se puede observar que se ha obtenido una precisión del 96,57% y sensibilidad (“recall”) del 97.69% para un threshold de confianza del 58,4%. El threshold para calcular la precisión y sensibilidad corresponde a la confianza mínima que ha de tener el modelo sobre cada *bounding box* como para considerarlas positiva y devolverla como resultado de la detección. El valor de threshold óptimo ha sido calculado internamente por AutoML. Los valores de evaluación son altos e indican que el modelo podría funcionar suficientemente bien.

Por lo tanto, el siguiente paso es analizar los resultados cuando se ejecuta el modelo con imágenes que el sistema no ha visto en la fase de entrenamiento. Para visualizar estos resultados se ha creado un código de Python que procesa la respuesta del modelo y dibuja *bounding boxes* en las coordenadas recibidas, usando un color diferente para las diferentes variedades de café. En las imágenes de la Figura 3.11 se pueden observar las fotografías originales a la izquierda y los resultados de la predicción a la derecha.

Como se confirma en la Figura 3.11, el modelo genera unos resultados muy buenos. Se puede observar cómo ha detectado correctamente el 100% de los productos de las tres imágenes presentadas. Además, la clasificación también ha sido correcta para todos los productos.

<sup>8</sup> El concepto precisión media, IoU, precisión y sensibilidad están definidos en el Anexo 1.



**Figura 3.11:** Imágenes originales a la izquierda, y las predicciones del primer modelo dibujadas a la derecha.

Por otro lado, también se ha podido notar en otras imágenes que hay algunas situaciones o productos con los que tiene más dificultad en detectar y clasificar cometiendo algún error. No obstante, las situaciones en las que falla son muy pocas comparadas con las situaciones en las que tiene éxito. En la Figura 3.12 se presenta una imagen, con los resultados del modelo dibujados, en la que comete un error de detección y uno de clasificación.



**Figura 3.12:** Imagen con la predicción del modelo. En esta imagen se produce un error de detección y otro de clasificación.

Los dos errores de la Figura 3.12 se muestran ampliados en las Figuras 3.13 y 3.14 para analizarlos más detalladamente. Por un lado, se puede ver como la imagen de la Figura 3.13 se produce un error de detección, no se ha marcado la caja L'OR que es parcialmente visible. Por otro lado, en la imagen de la figura 3.14, se ha clasificado con la misma clase dos facings distintos, ONYX\_20 y RISTRETTO\_20. Esto es visible ya que son cajas ligeramente distintas y se les ha asignado a todas el color verde que debería ser usado solamente para la variedad ONYX.



**Figura 3.13:** Error de detección en la imagen de la Figura 3.12.



**Figura 3.14:** Error de clasificación en la imagen de la Figura 3.12.

### 3.4 Creación del segundo conjunto de datos

Con la intención de mejorar el primer modelo generado e intentar solucionar los errores que comete, se procede a generar un segundo conjunto de datos, el cual estará compuesto por las imágenes de las que ya se disponía anteriormente más otras 60 nuevas, siendo estas 60 de puntos de venta donde aún no se habían hecho fotografías.

El proceso a seguir para generar este segundo conjunto de datos es parecido al proceso seguido para la creación del primero, descrito en la sección 3.1, pero con algunos ligeros cambios. El proceso se presenta a continuación:

1. Recolecta de imágenes, haciendo antes un análisis de las clases que tenían etiquetas y así intentar encontrar esos facings. Además, también se buscaron productos como PAPÚA NUEVA GUINEA o INDIA, los cuales no estuvieron en el primer conjunto de datos. La intención también es conseguir que el segundo modelo aprenda a clasificar entre más productos. Como se ha expuesto, se consiguen 60 imágenes más de las que se disponían en el primer conjunto de datos, siendo un total de 122 las que formarán parte del segundo *dataset*.
2. Usando otra vez LabelImg, se etiquetan las nuevas imágenes. Sin embargo, esta tarea se consiguió hacer más rápido con la ayuda del primer modelo que se generó anteriormente. Vistos los buenos resultados del primer modelo, se generó un código en Python que seguía el proceso que se muestra a continuación para cada una de las nuevas imágenes:
  - a. Leer la imagen.
  - b. Pasarla al primer modelo para recibir su respuesta.
  - c. Procesar la respuesta retornada.

- d. Crear un archivo XML con el mismo formato que genera el programa LabelImg y con la información de la predicción.

De esta forma, era posible abrir el programa etiquetador LabelImg con la mayoría de las *bounding boxes* ya situadas en el lugar adecuado y clasificadas correctamente, cosa que ahorró mucho trabajo de etiquetado. Sin embargo, la tarea de revisar una por una las *bounding boxes* creadas por el modelo era inevitable.

El resultado de este etiquetado fueron un total de 2627 *bounding boxes*.

3. Subir las imágenes a Google Cloud Storage.
4. Transformación de los datos para adaptarlos a AutoML, usando el mismo código de Python que se usó para la misma tarea con el primer conjunto de datos.
5. Importar el archivo CSV a AutoML.
6. Eliminar las etiquetas escasas. Esta vez, fué posible mantener 25 clases con las suficientes imágenes como para entrenar el segundo modelo. Sin embargo, al añadir fotos de productos que no se habían visto antes pero con pocos ejemplos, había más clases que se tuvieron que eliminar.

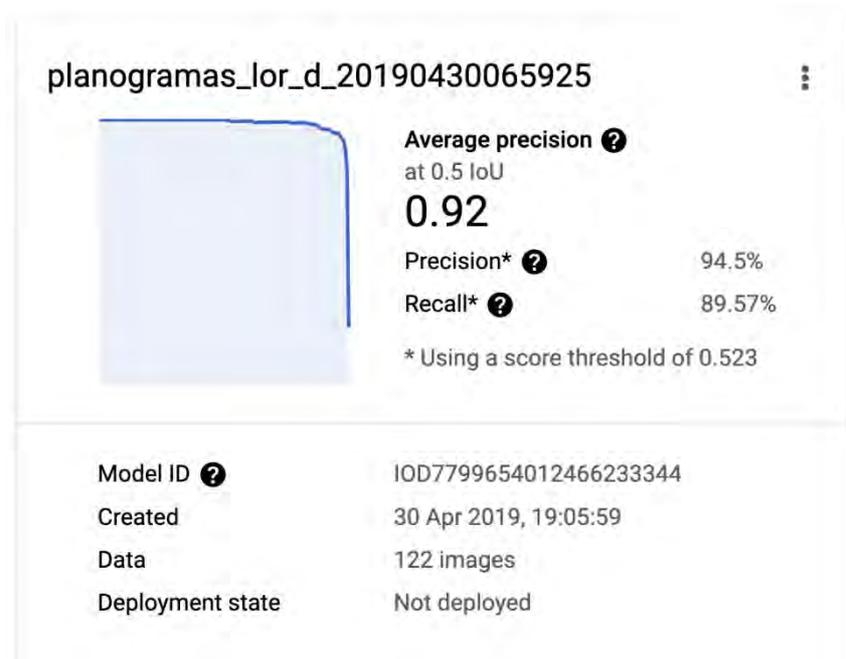
### 3.5 Segundo entrenamiento

Una vez ya está disponible el segundo conjunto de datos, el siguiente paso es entrenar el segundo modelo de AutoML Vision Object Detection. Tal y como se procedió para entrenar el primero en la sección 3.2, esta vez también optimizamos el modelo para darle más importancia a la exactitud antes que a la velocidad de respuesta.

Este segundo entrenamiento tardó, casualmente, también 2 horas y 44 minutos. Por consiguiente, también supone un coste aproximado de \$85.

### 3.6 Evaluación del segundo modelo

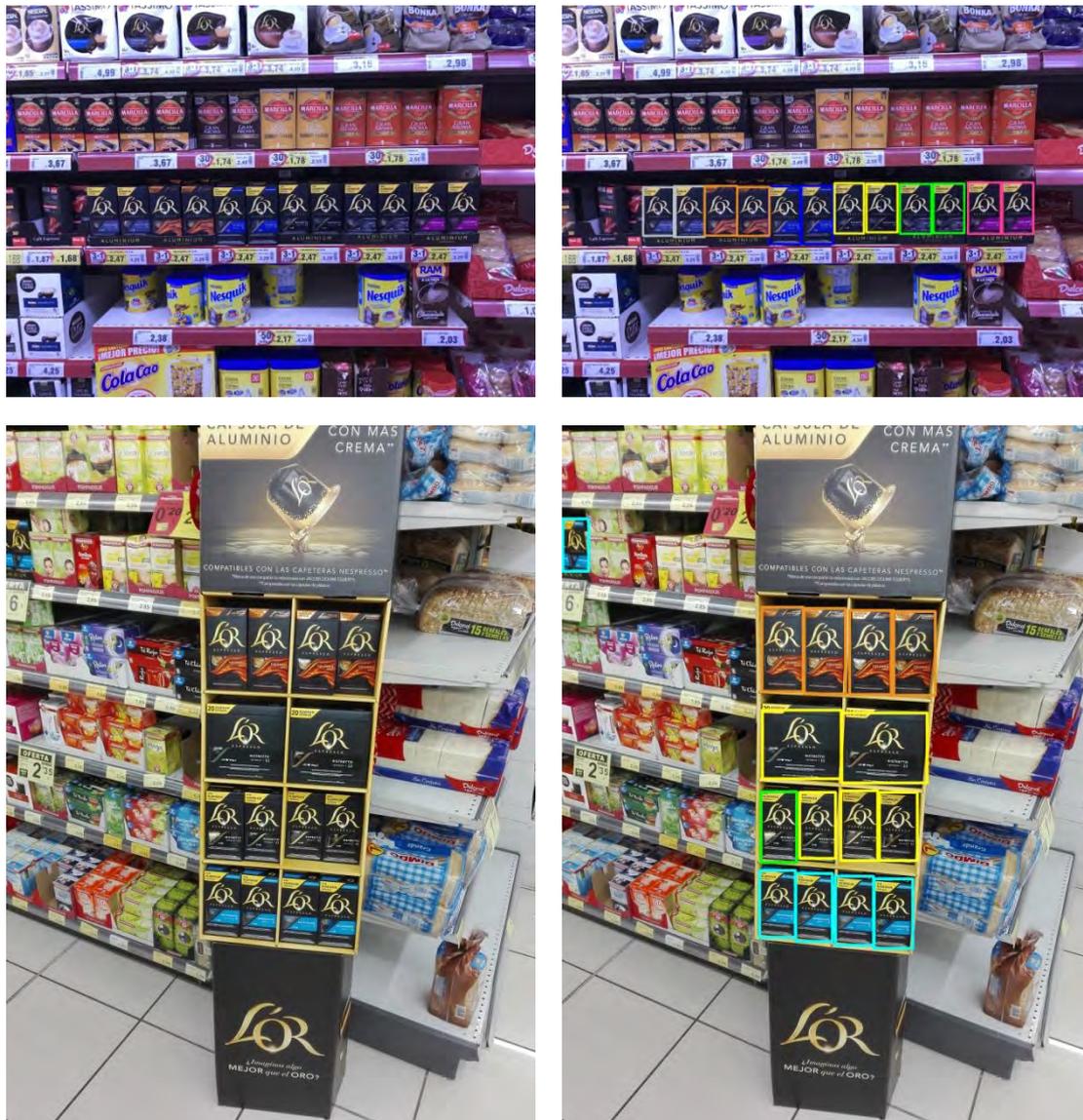
Con el segundo modelo ya entrenado, se analiza la evaluación que hace AutoML al mismo, usando las imágenes de “TEST” tal y como se ha detallado anteriormente. Para el nuevo modelo se obtiene 92% de precisión media usando un IoU del 50%. En este caso, la precisión es del 94,5% y sensibilidad del 89,57% para un threshold de confianza de 52,3%. Estos resultados se pueden observar en la Figura 3.15.



**Figura 3.15:** Evaluación de AutoML del segundo modelo.

Como se puede ver, este segundo modelo ha obtenido resultados ligeramente más bajos que el anterior según los números de la evaluación de AutoML. Esto puede ser debido a que, en el momento de suprimir una mayor cantidad de clases escasas del conjunto de datos entrenamiento, desaparecen también las *bounding boxes* de esa clase en las imágenes. De tal modo, se le está indicando que allí no hay ningún producto cuando realmente sí que lo hay. Y es que, de acuerdo con la documentación de Google, las imágenes deben ser exhaustivamente etiquetadas: si se quieren detectar coches, absolutamente todos los coches que aparezcan en una imagen de entrenamiento deberán ser etiquetados. Entonces, en este caso, en el proceso de entrenamiento el modelo entiende que se dan casos en los que no debe etiquetar los productos L'OR, causándole una leve confusión.

Siguiendo los mismos pasos que para el primer modelo, probamos con imágenes que el sistema no haya visto en la fase de entrenamiento. Los resultados, visiblemente muy correctos, se pueden ver en la Figura 3.16.



**Figura 3.16:** Imágenes originales a la izquierda, y las predicciones del segundo modelo dibujadas a la derecha.

Analizando los resultados de este segundo modelo, se puede afirmar que genera detecciones muy correctas. Incluso, en la imagen expuesta en la fila inferior, ha conseguido detectar el producto L'OR ESPRESSO DECAFFEINATO situado en el margen izquierdo de la imagen, producto que, tras hacer una ojeada rápida a la imagen, no se ve. No obstante, también produce algún error de clasificación y detección, pero cuesta dar con estos casos.

### 3.7 Comparación entre ambos modelos

Con tal de determinar cuál de los dos modelos será usado para verificar el cumplimiento de planogramas en nuestro sistema final, procedemos a realizar una comparación entre ambos.

Si comparamos cuantitativamente los dos modelos de detección y clasificación de facings de la marca L'OR, el primer modelo sale ganando en cuanto a los precisión media, precisión y sensibilidad. En la siguiente Tabla 3.3 se muestra una comparación entre ambos.

|                | Precisión media | Precisión usando el threshold óptimo | Sensibilidad usando el threshold óptimo |
|----------------|-----------------|--------------------------------------|---|
| Primer modelo  | 98,7 %          | 96,57 %                              | 97,69 %                                 |
| Segundo modelo | 92 %            | 94,5 %                               | 89,57 %                                 |

**Tabla 3.3:** Precisión media, precisión y sensibilidad para el primer y segundo modelo.

Por otro lado, se comparan cualitativamente ambos modelos. Para ello se pasan un conjunto de imágenes por cada uno de ellos y se visualizan las predicciones. En la Figura 3.17 y 3.18 se pueden analizar estos resultados.

Como se puede observar en la Figura 3.17, el segundo modelo ha sido capaz de detectar todos los facings, incluso los facings superiores de las cajas de la variedad SONTUOSO, cosa que el primer modelo no ha conseguido hacer. Además, también detecta y clasifica correctamente aquellos facings que nos causan problema en el primer modelo en la imagen de la Figura 3.18, la misma que se ha presentado en la sección 3.3. Y es que, por lo general, se observa el mismo comportamiento en todas las imágenes, una detección y clasificación más precisa en el segundo modelo.

Por otro lado, también se puede notar que las *bounding boxes* que se dibujan con el primer modelo no se ajustan tanto al facing real como sí lo hace el segundo modelo. Esto se puede notar especialmente en los productos de la zona superior de la Figura 3.18.

Es importante destacar que no ha sido necesario eliminar para ninguno de los modelos aquellas *bounding boxes* marcadas como “difficult” o “truncated”, ya que los resultados han sido muy buenos usándolas. Incluso, se puede observar que el modelo final es capaz de detectar algunos facings en situaciones complicadas, realizando una mejor actuación de la que se esperaba conseguir en un principio.

Finalmente se puede decir que, aunque los números del segundo modelo sean ligeramente más bajos que los del primero, los resultados del último parecen mejores al ser expuestos en imágenes. Tras llegar a estas conclusiones, se decide optar por el segundo modelo para utilizarlo como responsable de la detección de los facings de las imágenes que suban los GPVs al sistema.



*Figura 3.17: Resultados del primer (izquierda) y segundo (derecha) modelo en la imagen de un expositor.*



*Figura 3.18: Resultados del primer (izquierda) y segundo (derecha) modelo en la imagen de un lineal.*



## 4. DESARROLLO DE LA APLICACIÓN

Con la intención de ofrecer la mejor solución posible para realizar todo el proceso de verificación de cumplimiento de planogramas, se decide utilizar las últimas tecnologías para la creación de la aplicación web. En este capítulo se detallan desde los lenguajes y tecnologías usadas para que esta funcione en su totalidad, hasta el diseño y las funciones que tiene cada una de las pantallas.

### 4.1 Diseño

Analizando las funciones que se deben poder realizar desde la aplicación, recogidas en la sección 1.3 de este trabajo, ya es posible desarrollar el diseño de estas pantallas para cubrir los requerimientos.

#### 4.1.1 Material UI

Con el fin de desarrollar una aplicación con la mejor experiencia de usuario posible y un diseño agradable pero sin tener que dedicar mucho esfuerzo al aspecto de los elementos que aparecen en la pantalla, se usa Material UI. Material UI es una librería muy conocida de ReactJS (tecnología en la que se basa la parte visual y lógica de la aplicación y que se detalla en la sección 4.5.4 de este capítulo) que nos ofrece una gran cantidad de componentes ya implementados con el diseño de Google Material Design<sup>9</sup>.

Google Material Design ofrece muchos recursos gráficos para los desarrolladores: desde principios de diseño a seguir para ofrecer una buena experiencia de usuario, hasta imágenes para iconos y colores con los que combinar nuestros elementos.

#### 4.1.2 Descripción por pantallas

En este punto se listan y se describen las características de cada una de las pantallas. Es importante destacar que, durante esta descripción, las ventanas emergentes serán consideradas pantallas diferentes para hacer más cómoda la explicación y comprensión del funcionamiento.

---

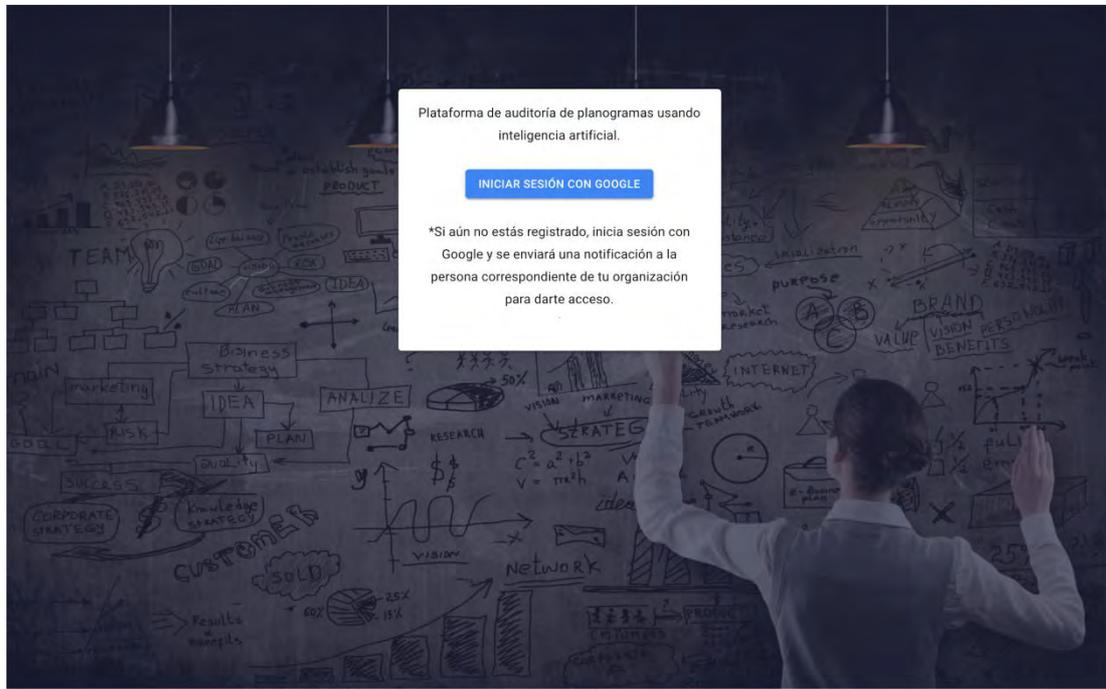
<sup>9</sup> Página oficial de Google Material Design: <https://material.io>

## Pantalla A

NOMBRE

Iniciar sesión

CAPTURA DE LA PANTALLA



*Figura 4.1: Pantalla A. Iniciar sesión.*

DESCRIPCIÓN

Pantalla desde la cual los usuarios inician sesión. Siempre será usando una cuenta de Google. Al pulsar INICIAR SESIÓN CON GOOGLE los usuarios son redirigidos a la página común de autenticación de Google desde la cual se selecciona el usuario para acceder.

Aquellas personas que entran a la plataforma y no tienen permiso no podrán acceder hasta que un usuario con el rol de Trade Marketing le permita el acceso desde la pantalla de Gestión de Usuarios (pantalla P).

## Pantalla B

### NOMBRE

Histórico de visitas

### CAPTURA DE LA PANTALLA

| Fecha ↓    | Hora  | Punto de venta | Cumplimiento de productos | Cumplimiento de área |
|------------|-------|----------------|---------------------------|----------------------|
| 10/06/2019 | 13:52 | Alcampo        | 57 %                      | 68 %                 |
| 10/06/2019 | 14:25 | plusfrut       | 0 %                       | 175 %                |
| 10/06/2019 | 14:07 | caprabo        | 100 %                     | 237 %                |
| 10/06/2019 | 14:13 | caprabo        | 75 %                      | 250 %                |
| 10/06/2019 | 14:17 | caprabo        | 75 %                      | 250 %                |
| 04/06/2019 | 14:49 | MERCADONA      | 11 %                      | 28 %                 |
| 04/06/2019 | 15:07 | MERCADONA      | 61 %                      | 142 %                |
| 04/06/2019 | 16:45 | MERCADONA      | 46 %                      | 118 %                |
| 03/06/2019 | 18:05 | MERCADONA      | 0 %                       | 0 %                  |
| 01/06/2019 | 14:46 | Dia            | 100 %                     | 162 %                |

*Figura 4.2: Pantalla B. Histórico de visitas.*

### DESCRIPCIÓN

Esta pantalla está compuesta por una tabla en la cual se muestra, por filas, una lista de las últimas visitas hechas a los distintos puntos de venta. La lista, que está ordenada por fecha y hora, muestra también el logo o nombre del punto de venta que se ha visitado y los dos porcentajes de cumplimiento para cada una de las visitas.

Es importante destacar las herramientas que ofrece la tabla en la barra superior. De izquierda a derecha, estas son:

- Buscar una visita concreta. Haciendo clic en la lupa es posible introducir cualquier texto, para buscar una visita concreta en función del valor de cualquiera de las columnas.
- Descargar un archivo CSV con los datos que se muestran en la tabla, preparado para ser introducido en cualquier hoja de cálculo.
- Imprimir la tabla.
- Seleccionar qué columnas se desean visualizar en la tabla.
- Aplicar filtros para mostrar únicamente aquellas visitas que los cumplan.

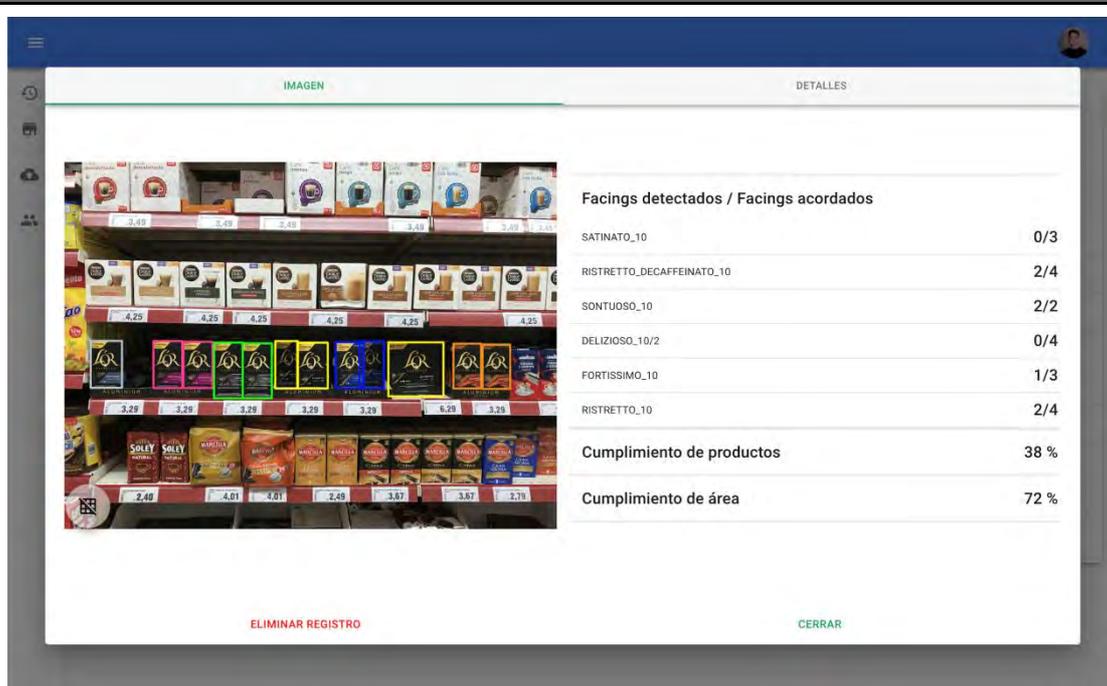
Haciendo clic en cualquiera de las filas, se abre una pantalla emergente (pantalla C) con toda la información disponible sobre la visita seleccionada.

## Pantalla C (ventana emergente)

### NOMBRE

Imagen de la visita

### CAPTURA DE LA PANTALLA



*Figura 4.3: Pantalla C. Imagen de la visita.*

### DESCRIPCIÓN

Esta ventana emergente tiene la principal función de mostrar la imagen del lineal expositor que ha tomado el GPV, combinada con los resultados de la predicción del modelo de inteligencia artificial. En la zona izquierda se observa esta imagen con recuadros dibujados en la zona que el modelo ha detectado productos de la marca. Estos recuadros se pueden esconder haciendo uso del botón que se encuentra en las esquina inferior izquierda de la imagen. En la zona derecha se muestra para cada facing, la cantidad que ha detectado el modelo y la cantidad que debería haber realmente según el planograma. También se enseñan aquí los dos porcentajes de cumplimiento.

Haciendo clic en la barra superior de esta ventana emergente (o deslizando en caso de estar usando un ordenador con pantalla táctil) es posible navegar entre las pantallas C y D. Para cerrarla es necesario pulsar el botón CERRAR o hacer clic fuera de la superficie de la ventana.

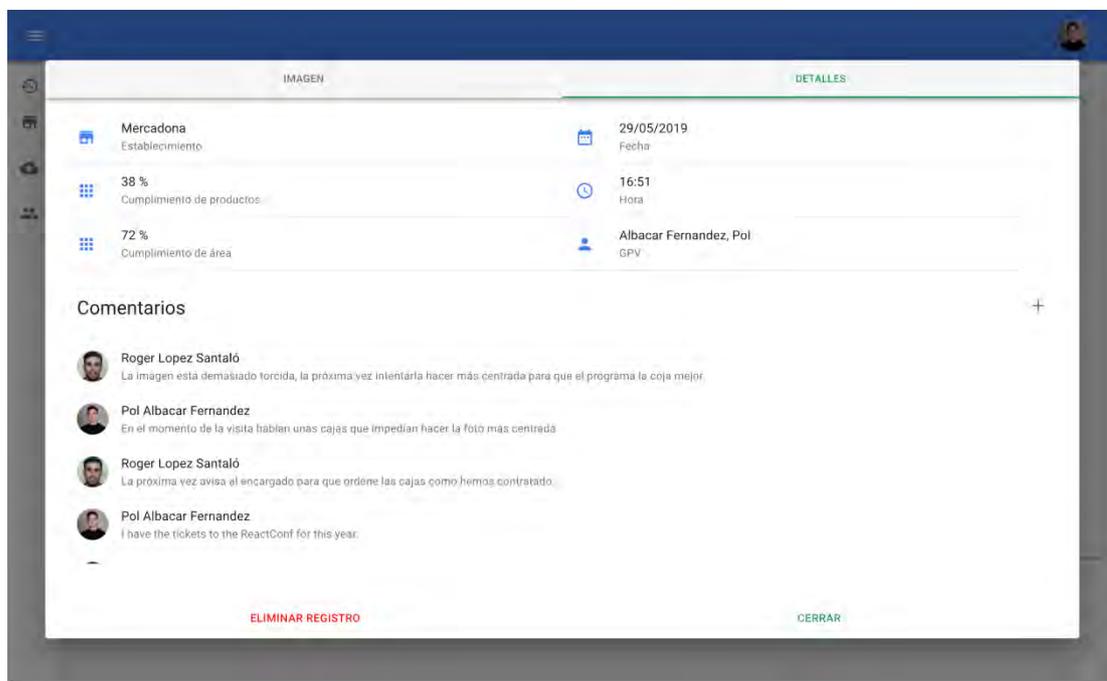
Además, en la esquina inferior izquierda se tiene la posibilidad de eliminar la visita.

## Pantalla D (ventana emergente)

### NOMBRE

Detalles de la visita

### CAPTURA DE LA PANTALLA



**Figura 4.4:** Pantalla D. Detalles de la visita.

### DESCRIPCIÓN

Desde aquí se visualiza más información de la visita realizada: como el GPV que ha realizado la visita, fecha y hora y nombre del establecimiento. Además, se ha añadido la funcionalidad de añadir comentarios al registro para poder matizar la visita, en caso de encontrar algo fuera de lo normal. Para ello, hay que pulsar el icono con el símbolo (+) que hay en la zona derecha de la ventana.

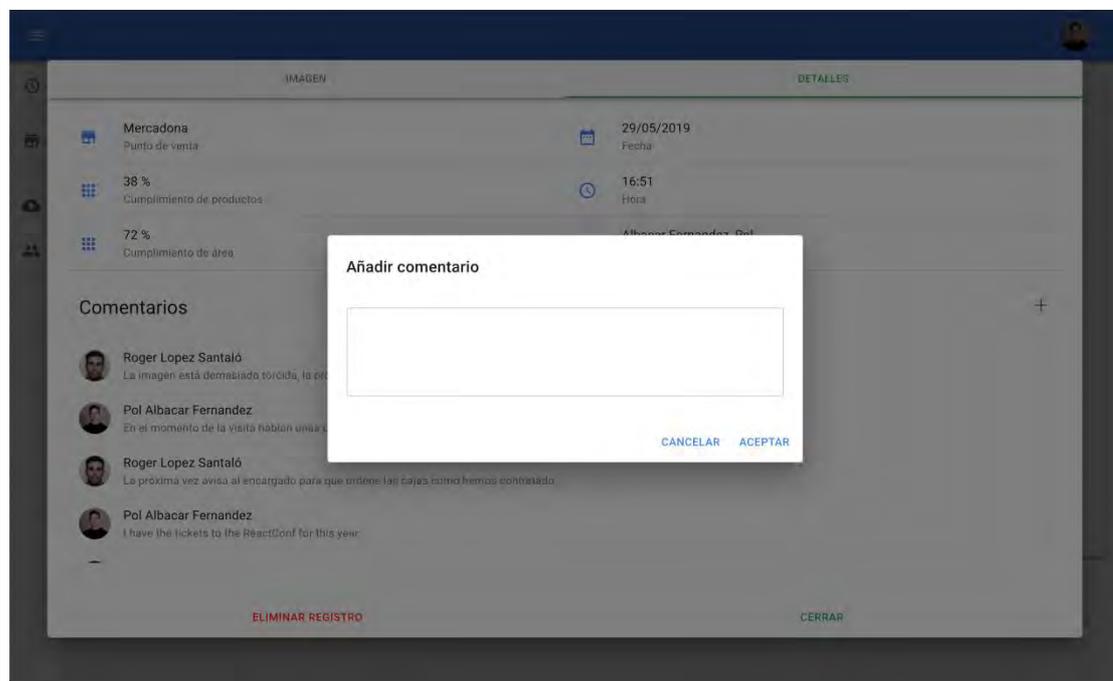
Igual que en la pantalla C, haciendo clic en la barra superior es posible navegar entre las pantallas C y D, cerrar la ventana y también eliminar la visita de igual forma.

Pantalla E (ventana emergente):

**NOMBRE**

Añadir comentario a la visita

**CAPTURA DE LA PANTALLA**



*Figura 4.5: Pantalla E. Añadir comentario a la visita.*

**DESCRIPCIÓN**

Tal y como el nombre de la pantalla indica, desde aquí se añaden los comentarios a la visita que ha realizado el GPV. Los comentarios se añaden para poder especificar, por ejemplo, el porqué se ha hecho una imagen desde lejos.

## Pantalla F:

### NOMBRE

Puntos de venta

### CAPTURA DE LA PANTALLA

| ID   | Nombre    | Población       | Provincia | Cumplimiento de productos | Cumplimiento de área | Mejora  |
|------|-----------|-----------------|-----------|---------------------------|----------------------|---------|
| 0001 | MERCADONA | Vilassar de Mar | Barcelona | 46 %                      | 118 %                | -19 % ↓ |
| 0002 | Carrefour | Cabrera de Mar  | Barcelona |                           |                      |         |
| 0003 | Alcampo   | Mataró          | Barcelona | 57 %                      | 68 %                 | 35 % ↑  |
| 0004 | plusfrésc | Barcelona       | Barcelona | 0 %                       | 175 %                |         |
| 0005 | caprabo   | Mataró          | Barcelona | 100 %                     | 125 %                | 0 %     |
| 0006 | caprabo   | Barcelona       | Barcelona | 100 %                     | 237 %                | 93 % ↑  |
| 0007 | caprabo   | Mataró          | Mataró    | 75 %                      | 250 %                | 0 %     |
| 0008 | caprabo   | Mataró          | Barcelona | 100 %                     | 221 %                |         |
| 0009 | Condis    | Mataró          | Barcelona | 100 %                     | 225 %                |         |
| 0010 | Condis    | Mataró          | Barcelona | 87 %                      | 337 %                |         |
| 0011 | e consum  | Mataró          | Barcelona | 100 %                     | 375 %                | 37 % ↑  |
| 0012 | e consum  | Mataró          | Barcelona | 100 %                     | 375 %                |         |
| 0013 | Dia       | Mataró          | Barcelona | 100 %                     | 162 %                |         |

Figura 4.6: Pantalla F. Puntos de venta.

### DESCRIPCIÓN

Muestra una tabla completa donde cada fila corresponde a cada uno de los puntos de venta en los cuales la marca tiene productos expuestos. Desde esta tabla es posible buscar un comercio concreto y acceder a información más detallada haciendo click encima. En cada fila, además, se muestra información básica del resultado de la última visita realizada, así como los dos porcentajes de cumplimiento y la mejora respecto a la anterior visita.

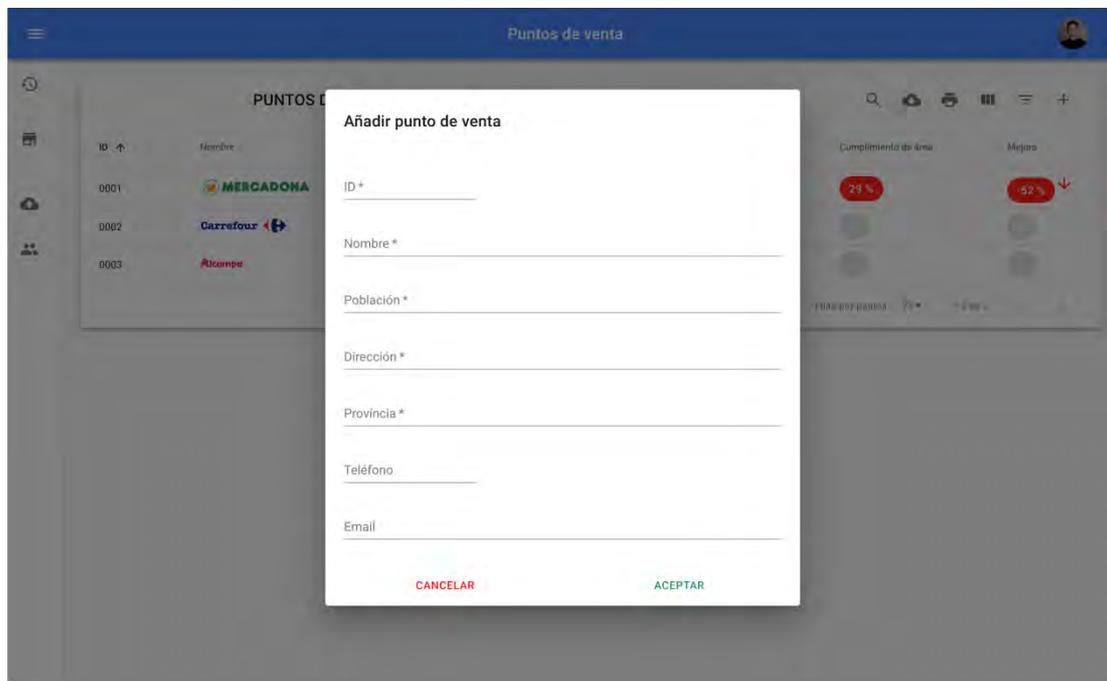
En la barra superior de la tabla, se ofrecen las mismas herramientas que se encuentran en la tabla de la pantalla C, pero se añade la posibilidad de crear nuevos puntos de venta haciendo clic en el icono (+).

Pantalla G (ventana emergente):

**NOMBRE**

Añadir nuevo punto de venta

**CAPTURA DE LA PANTALLA**



*Figura 4.7: Pantalla G. Añadir nuevo punto de venta.*

**DESCRIPCIÓN**

Ventana desde la cual se crean puntos de venta. Para ello, es necesario introducir los datos de:

- ID: valor que selecciona la persona que crea el punto de venta. Debe ser único.
- Nombre del punto de venta
- Población en la que se encuentra el punto de venta
- Dirección exacta
- Provincia
- Teléfono de contacto
- Email de contacto

Todos los datos se requieren para poder crear el punto de venta, excepto el teléfono y email de contacto.

Pantalla H:

NOMBRE

Punto de venta

CAPTURA DE LA PANTALLA

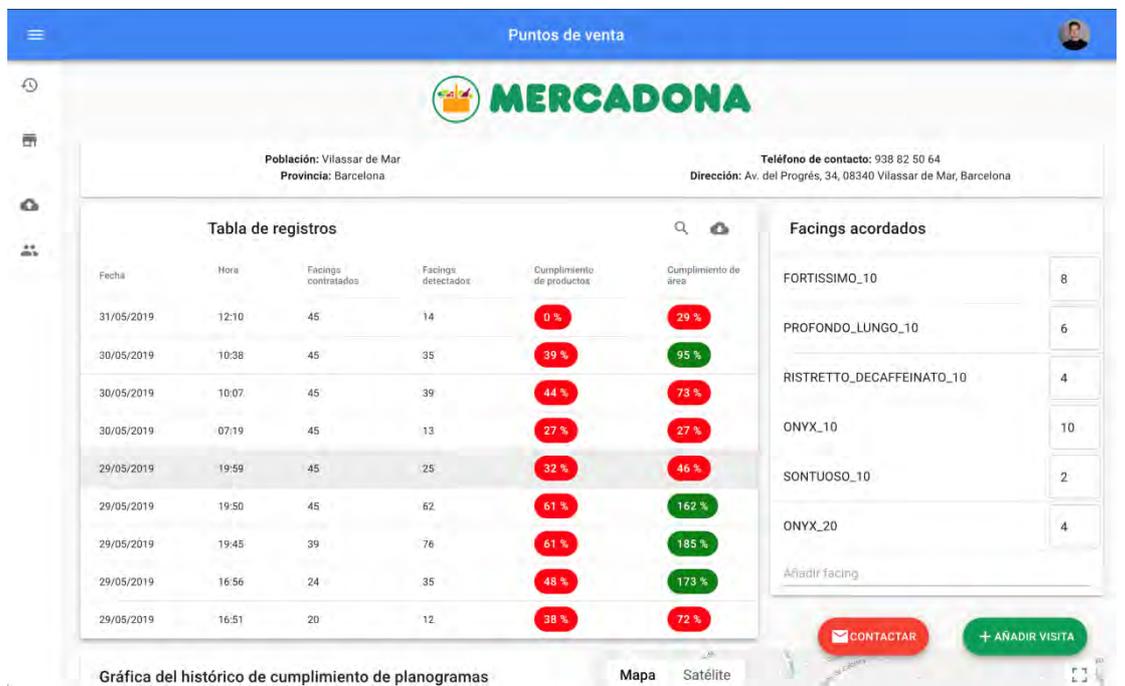


Figura 4.8: Pantalla H. Punto de venta (1).

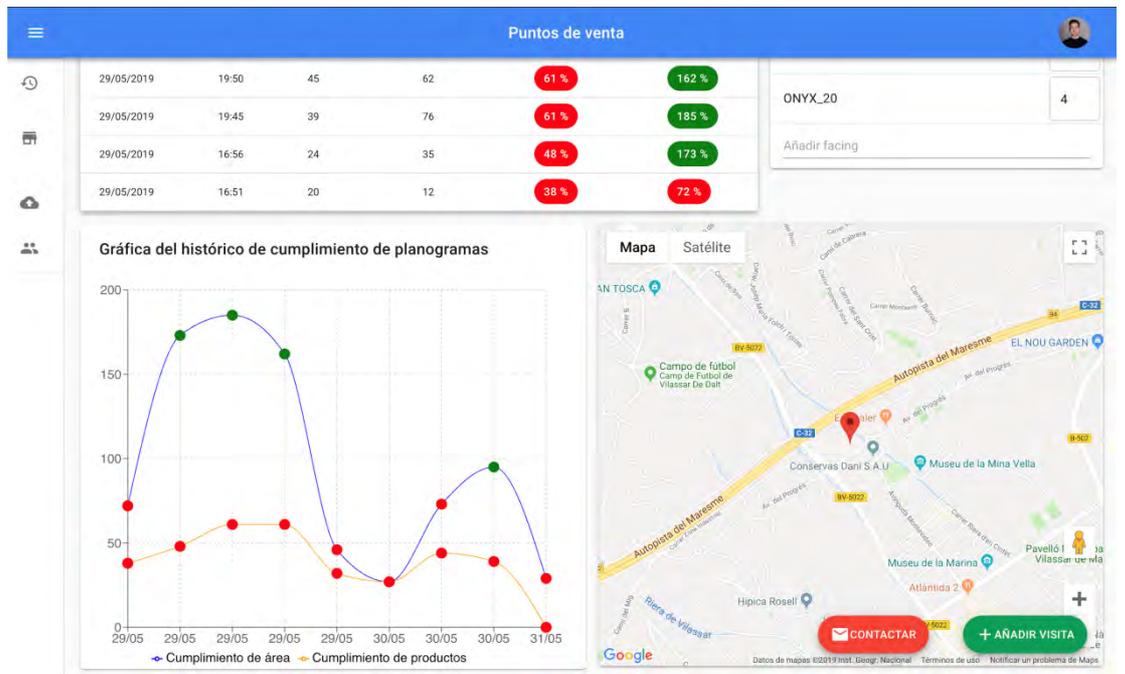


Figura 4.9: Pantalla H. Punto de venta (2).

## DESCRIPCIÓN

Pantalla en la cual se muestra todo tipo de información sobre un punto de venta concreto.

Primeramente, en la zona superior, se muestran el nombre y datos básicos, como son la población, provincia, dirección y teléfono de contacto.

El contenido restante se divide en 4 zonas diferenciadas.

- Zona superior-izquierda: tabla con la lista de las últimas visitas que se han realizado en ese punto de venta, ordenadas por fecha y hora. Igual que en la pantalla B, en cada fila se observa información de cada visita. Al seleccionar cualquiera de las visitas, se despliega la pantalla C. Adicionalmente, también existe la posibilidad de buscar en la tabla y de descargar sus datos en formato CSV.
- Zona superior-derecha: lista de los facings acordados en el comercio seleccionado. Aquí es posible modificarlos, añadiendo facings y cambiando la cantidad de cada uno de ellos, según los acordados en el planograma. Una vez cambiados, es necesario guardar la acción mediante un botón que aparece en la pantalla con tal de confirmar los cambios.
- Zona inferior-izquierda: gráfica que muestra la evolución de los dos porcentajes de cumplimiento del planograma en las últimas visitas.
- Zona inferior-derecha: mapa con la posición exacta del punto de venta.

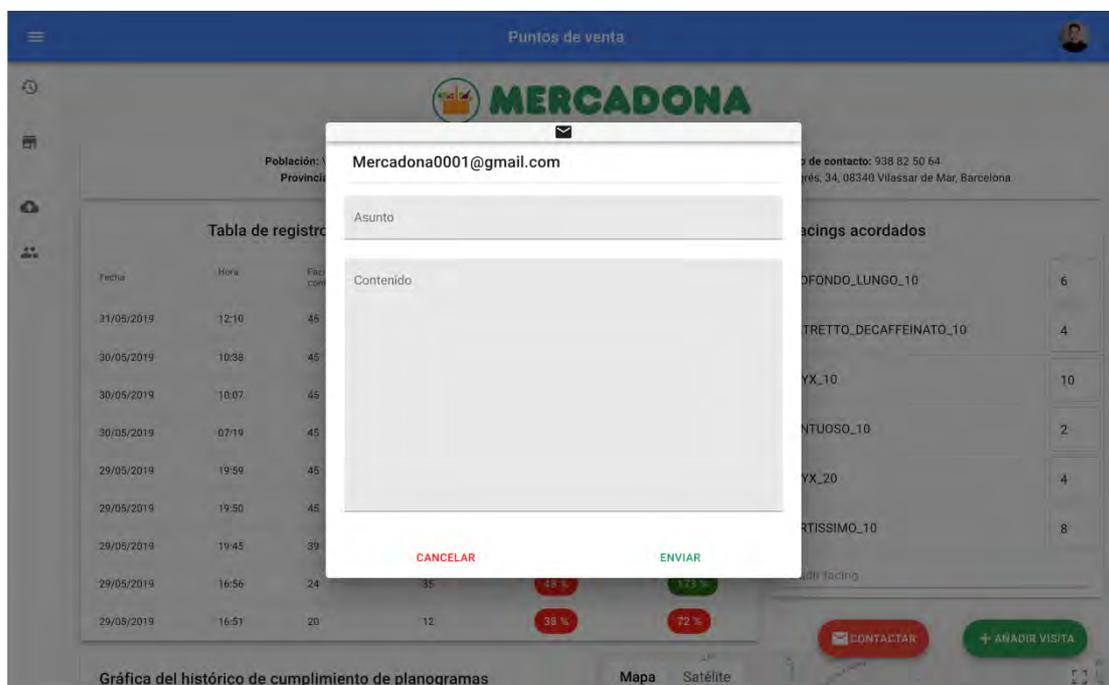
En la esquina inferior derecha se sitúan dos botones flotantes. El botón rojo con el texto CONTACTAR para desplegar la pantalla I desde donde se permite contactar por correo al punto de venta; y el botón verde con el texto AÑADIR VISITA para redirigir al usuario a la pantalla K.

Pantalla I (ventana emergente):

**NOMBRE**

Enviar correo electrónico al punto de venta

**CAPTURA DE LA PANTALLA**



*Figura 4.10: Pantalla I. Enviar correo electrónico al punto de venta.*

**DESCRIPCIÓN**

Pantalla útil para contactar con el punto de venta vía correo electrónico y comunicar cualquier asunto. Esta opción está disponible para aquellos comercios que se les ha añadido una dirección email. Los dos botones que contiene la ventana son CANCELAR y ENVIAR.

## Pantalla J:

### NOMBRE

Nueva visita, datos del punto de venta

### CAPTURA DE LA PANTALLA

The screenshot shows a mobile application interface for registering a new visit. At the top, a blue header contains the text 'Registrar una nueva visita' and a user profile icon. Below the header, a progress indicator shows three steps: 'Datos del establecimiento', 'Fecha y hora', and 'Subir imagen'. The current step is 'Datos del punto de venta', which includes a sub-header 'Introducir ID'. The form contains several input fields: 'Identificador de punto de venta', a radio button for 'o dirección del punto de venta', 'Nombre', 'Población', 'Dirección', 'Provincia', and 'País'. At the bottom, there is a checkbox labeled 'Añadir como nuevo punto de venta' and a blue button labeled 'SIGUIENTE'.

**Figura 4.11:** Pantalla J. Nueva visita, datos del punto de venta.

### DESCRIPCIÓN

Es la primera parte de un conjunto de tres pantallas que se deben completar para registrar una nueva visita. En la zona superior se observa el progreso realizado a través de una secuencia de pasos numerados.

En esta pantalla se introducen los datos del punto de venta en el cual se ha realizado la visita. Para completarlo y pasar al siguiente paso hay dos opciones: la primera es introducir el identificador del comercio, el cual se ha determinado en el momento de la creación; la segunda opción es rellenar los datos de la dirección del comercio. Además, también hay la posibilidad de registrar los datos introducidos como un nuevo punto de venta, marcando la caja que aparece en la zona inferior.

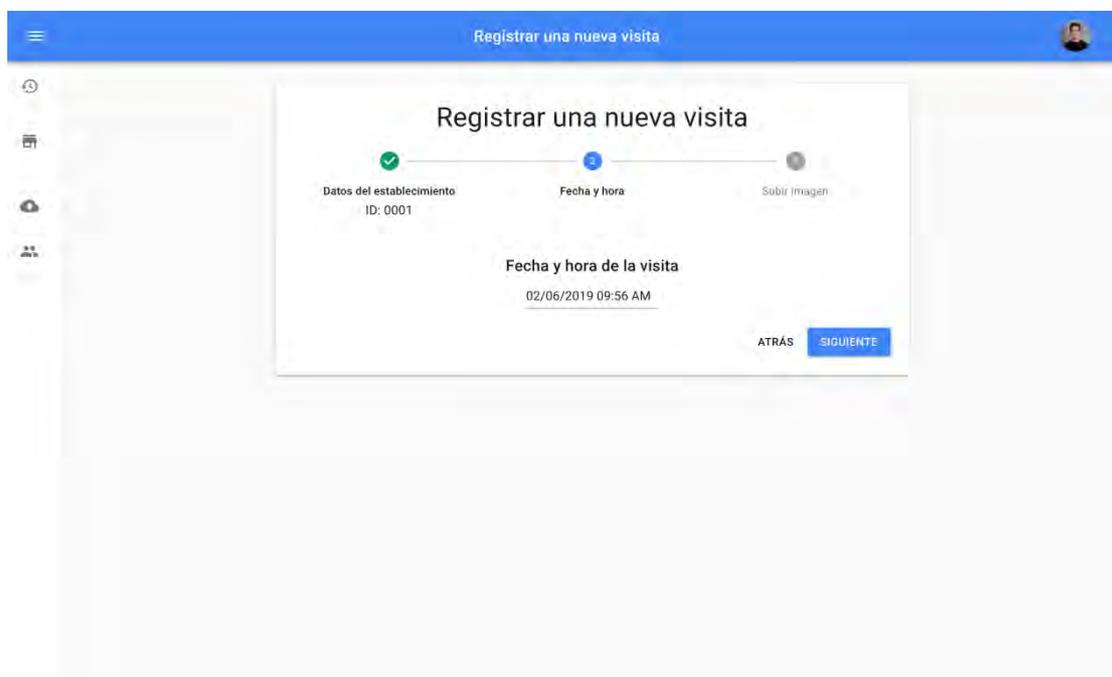
El botón SIGUIENTE, que se encuentra en la esquina inferior derecha del elemento principal, habilita pasar al siguiente paso: pantalla K.

Pantalla K:

**NOMBRE**

Nueva visita, fecha y hora de la visita.

**CAPTURA DE LA PANTALLA**



*Figura 4.12: Pantalla K. Nueva visita, fecha y hora de la visita.*

**DESCRIPCIÓN**

En esta pantalla se selecciona la fecha y hora en que se ha realizado la visita que se está registrando. Por defecto, la aplicación escribe la fecha y hora actual. Para cambiar la fecha y hora, hay que pulsar en el texto donde esta se muestra, es entonces cuando se abre una ventana emergente, pantalla L.

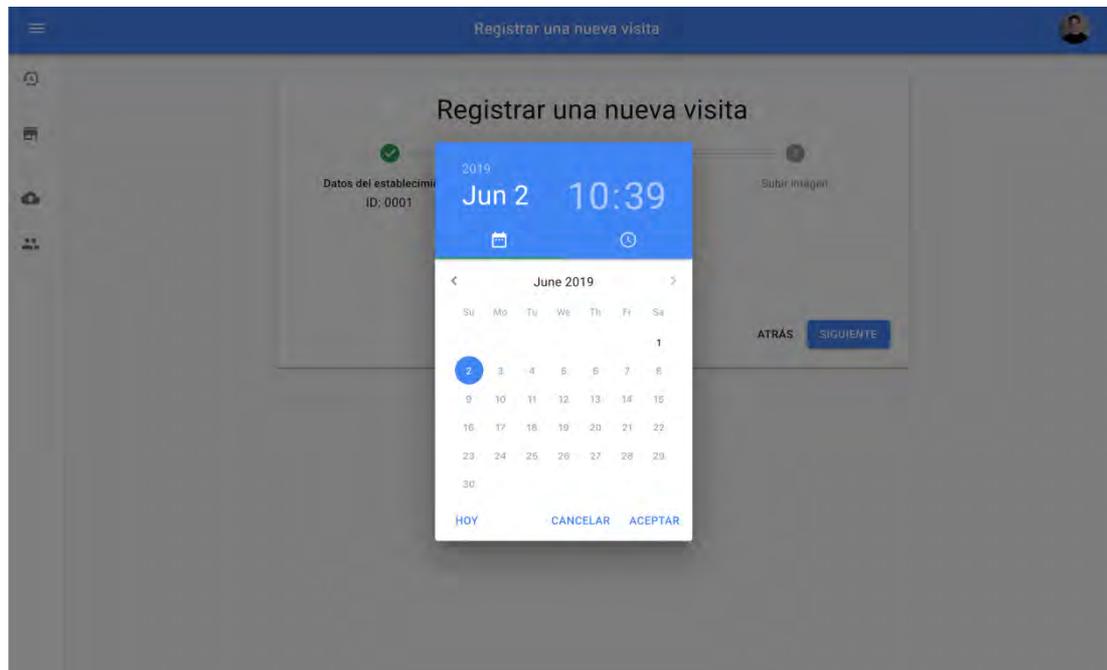
Cuando la fecha y hora son correctas, se pulsa SIGUIENTE para pasar al último paso, o ATRÁS para retroceder.

Pantalla L (ventana emergente):

**NOMBRE**

Nueva visita, selección de la fecha de la visita.

**CAPTURA DE LA PANTALLA**



**Figura 4.13:** Pantalla L. Nueva visita, selección de la fecha de la visita.

**DESCRIPCIÓN**

Ventana emergente desde donde se cambia de la fecha de la visita, navegando por el calendario y haciendo clic en cualquiera de los días pasados. Las fechas futuras han sido deshabilitadas para evitar que se creen visitas que realmente no se han realizado. Cuando se ha terminado, se confirma el cambio pulsando ACEPTAR.

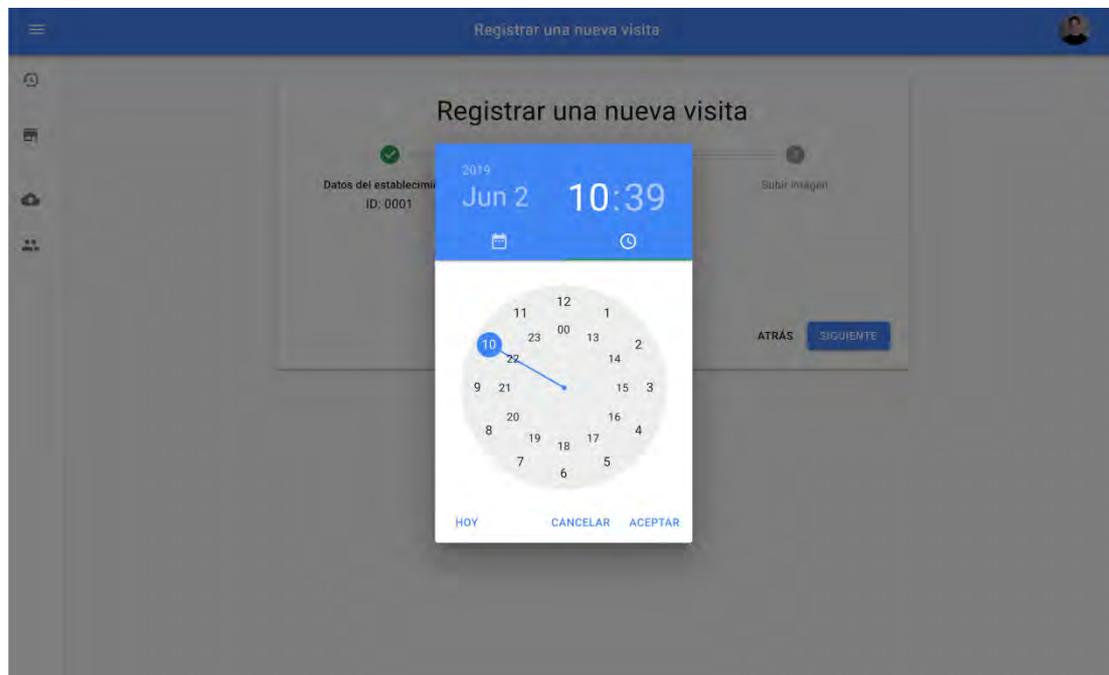
Haciendo clic en el reloj, situado arriba a la derecha, se procede a cambiar la hora de la visita, pantalla M. Si se presiona CANCELAR, o fuera de los dominios de la ventana emergente, vuelve a la pantalla K sin guardar los cambios.

Pantalla M (ventana emergente):

**NOMBRE**

Nueva visita, selección de la hora de la visita.

**CAPTURA DE LA PANTALLA**



**Figura 4.14:** Pantalla M. Nueva visita, selección de la hora de la visita.

**DESCRIPCIÓN**

Pantalla emergente desde donde se cambia de la hora de la visita, seleccionando primero la hora y seguidamente los minutos. Cuando se ha terminado, se confirma el cambio pulsando ACEPTAR.

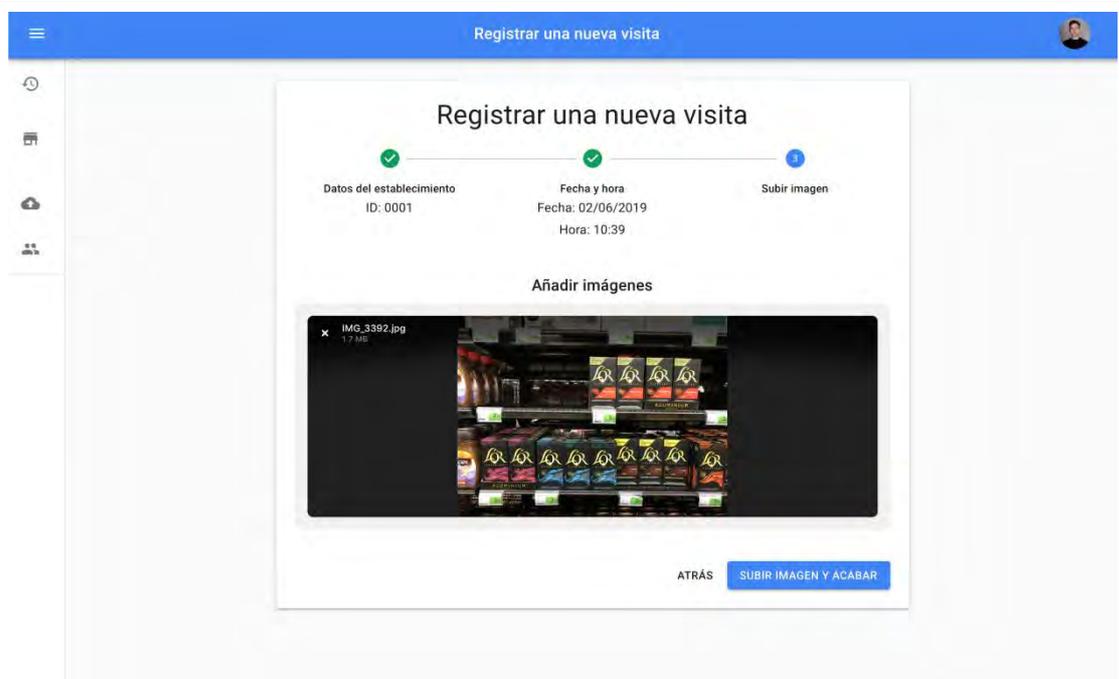
Haciendo clic en el icono que muestra un calendario, situado arriba a la izquierda, se retrocede a la selección de la fecha, pantalla L. También, igual que en la pantalla L, es posible presionar ACEPTAR o CANCELAR para salir guardando o sin guardar los cambios.

Pantalla N:

**NOMBRE**

Nueva visita, subir imagen.

**CAPTURA DE LA PANTALLA**



*Figura 4.15: Pantalla N. Nueva visita, subir imagen.*

**DESCRIPCIÓN**

Pantalla desde la cual se sube la imagen del lineal o expositor que se ha tomado en el punto de venta. La imagen se arrastra al área gris que aparece en el centro de la pantalla y seguidamente se previsualiza en la interfaz. Aquí se puede ver información del archivo que se ha subido, concretamente el nombre y el tamaño.

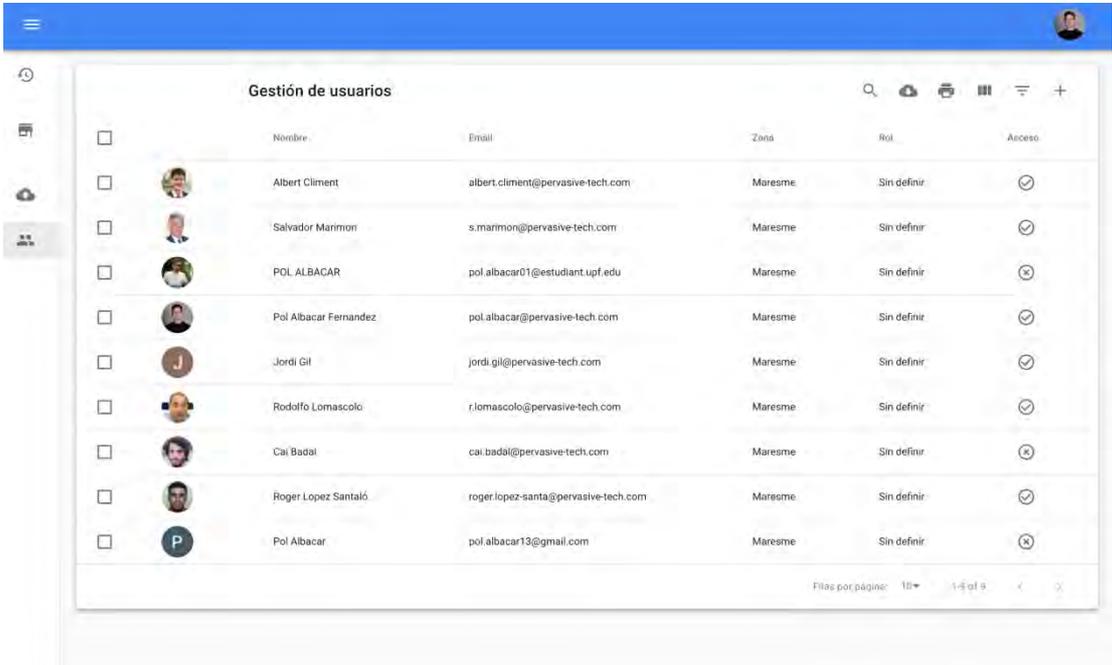
Para finalizar el registro de la visita, solamente es necesario presionar el botón SUBIR IMAGEN Y ACABAR, cosa que desencadenará el proceso de subir la imagen y aparecerá una barra de progreso. Cuando ha terminado, la aplicación redirige al usuario directamente a la página del punto de venta al cual se ha añadido la visita.

## Pantalla O:

### NOMBRE

### Gestión de usuarios

### CAPTURA DE LA PANTALLA



The screenshot shows a web application interface for user management. The title is 'Gestión de usuarios'. It features a table with the following columns: 'Nombre', 'Email', 'Zona', 'Rol', and 'Acceso'. Each row represents a user and includes a checkbox on the left and a status icon (checkmark or cross) in the 'Acceso' column. The table contains 10 rows of user data.

|                          | Nombre                | Email                                | Zona    | Rol         | Acceso |
|--------------------------|-----------------------|--------------------------------------|---------|-------------|--------|
| <input type="checkbox"/> | Albert Climent        | albert.climent@pervasive-tech.com    | Maresme | Sin definir | ✓      |
| <input type="checkbox"/> | Salvador Marimon      | s.marimon@pervasive-tech.com         | Maresme | Sin definir | ✓      |
| <input type="checkbox"/> | POL ALBACAR           | pol.albacar01@estudiant.upf.edu      | Maresme | Sin definir | ✗      |
| <input type="checkbox"/> | Pol Albacar Fernandez | pol.albacar@pervasive-tech.com       | Maresme | Sin definir | ✓      |
| <input type="checkbox"/> | Jordi Gil             | jordi.gil@pervasive-tech.com         | Maresme | Sin definir | ✓      |
| <input type="checkbox"/> | Rodolfo Lomascolo     | r.lomascolo@pervasive-tech.com       | Maresme | Sin definir | ✓      |
| <input type="checkbox"/> | Cai Badal             | cai.badal@pervasive-tech.com         | Maresme | Sin definir | ✗      |
| <input type="checkbox"/> | Roger Lopez Santaló   | roger.lopez-santa@pervasive-tech.com | Maresme | Sin definir | ✓      |
| <input type="checkbox"/> | Pol Albacar           | pol.albacar13@gmail.com              | Maresme | Sin definir | ✗      |

*Figura 4.16: Pantalla O. Gestión de usuarios.*

### DESCRIPCIÓN

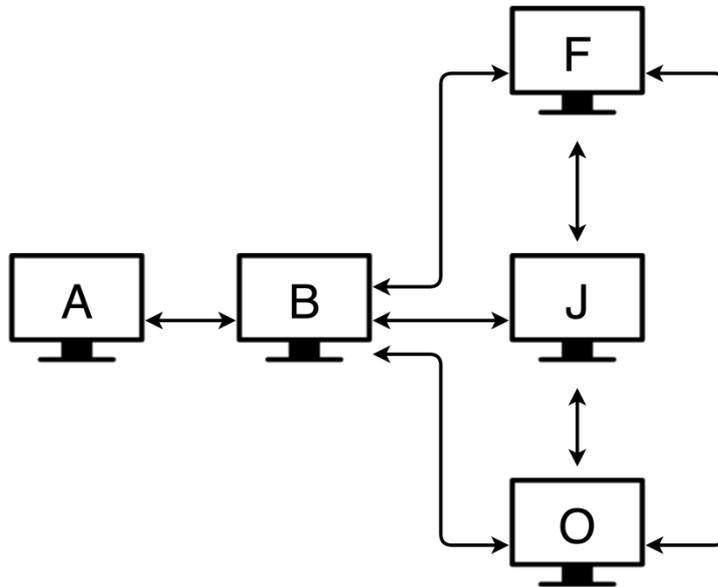
Desde esta pantalla es posible realizar una gestión de los usuarios que utilizan la aplicación. Se muestra una tabla con todos los usuarios que hay registrados en la base de datos, y para cada uno expone la foto de usuario de Google, nombre completo, email, zona asignada (zona donde realiza las visitas), rol y acceso.

No todas las personas que se registran en la aplicación tienen acceso a sus funcionalidades. Para disponer de acceso, una persona habilitada debe proporcionarlo desde esta pantalla. Para ello, simplemente se debe buscar el usuario en concreto al que se le quiere dar acceso y presionar la cruz que aparece en la última columna de la tabla. La cruz se convertirá en un símbolo de verificación.

Esta tabla también dispone de las herramientas que se ofrecen en las tablas de las demás pantallas.

### 4.1.3 Mapas de navegación

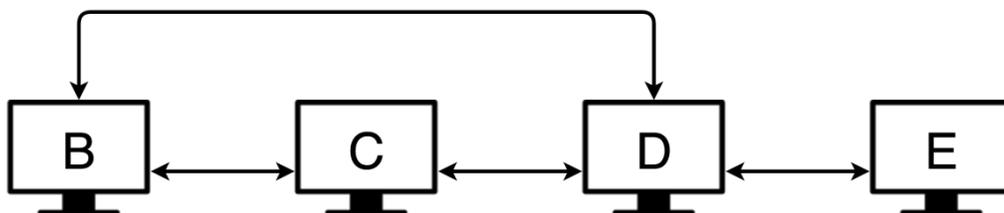
#### 4.1.3.1 Entre las pantallas principales



*Figura 4.17: Mapa de navegación entre las pantallas principales.*

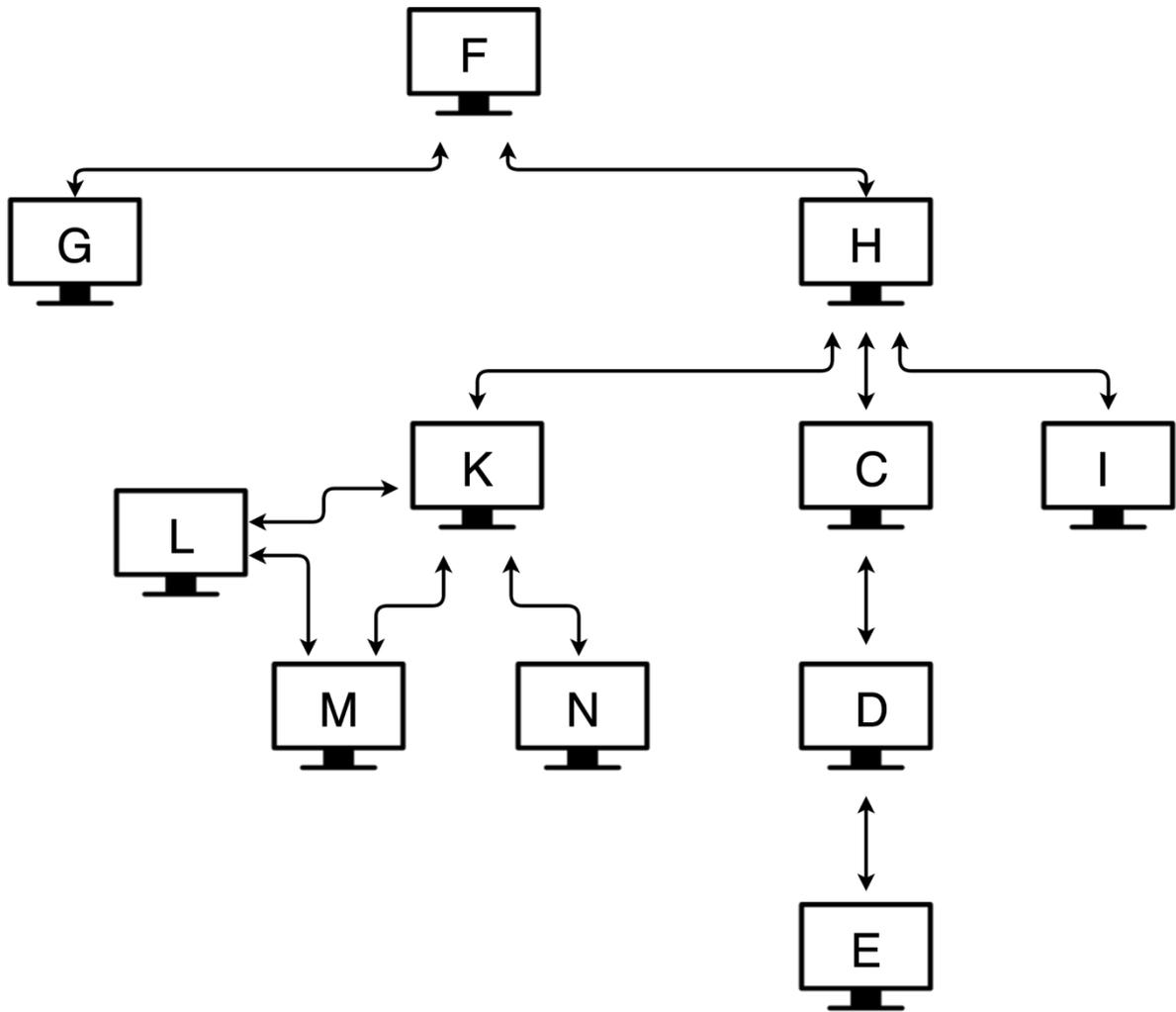
Es importante destacar que la navegación entre las principales pantallas se realizará mediante el menú desplegable que se encuentra en la esquina superior izquierda de la interfaz. También cabe destacar que la acción de cerrar la sesión se podrá realizar desde cualquier parte de la aplicación, haciendo clic en la imagen de usuario que aparece en la esquina superior derecha de la interfaz, y seguidamente haciendo clic en cerrar sesión.

#### 4.1.3.2 Desde la pantalla Histórico de visitas



*Figura 4.18: Mapa de navegación desde la pantalla Histórico de visitas.*

#### 4.1.3.3 Desde la pantalla Puntos de venta



*Figura 4.19: Mapa de navegación desde la pantalla Puntos de venta.*

#### 4.1.4 Avisos de errores y éxitos.

Para ofrecer una buena experiencia, es importante ofrecer respuestas para que el usuario pueda saber qué es lo que está pasando en la aplicación. Las respuestas que se deben tener en cuenta son los posibles errores que se puedan producir, la carga de datos o de alguna acción que necesita cierto tiempo y también las acciones que se realizan con éxito por parte del servidor y que el usuario debe conocer.

Es por eso que, para este proyecto, se han usado las barras de progreso para indicar que se está subiendo la imagen al servidor, y por otro lado círculos de progreso para indicar que se están cargando los datos. Estos se muestran en la Figura 4.20 y Figura 2.21, respectivamente.

**Figura 4.20:** Barra de progreso.

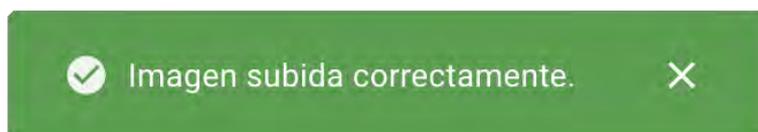


**Figura 4.21:** Círculo de progreso.

Por otra parte, se han utilizado los Snackbars de Material UI para indicar cuando ha habido un error, o cuando se ha producido una acción con éxito. En la Figura 4.22 y 4.23 podemos observar un ejemplo de cada. Estos aparecen en la esquina inferior izquierda.



**Figura 4.22:** Snackbar de error.



**Figura 4.23:** Snackbar de éxito.

## 4.2 Cliente-Servidor

La arquitectura cliente-servidor es un modelo de diseño de software organizado en dos partes claramente diferenciadas, por un lado el conjunto de servidores que realizan y ofrecen servicios, y por otro lado los clientes que acceden y usan los servicios de los servidores mediante una petición a través de internet. De tal modo, esta arquitectura permite repartir la capacidad de procesamiento entre ambas partes, ofreciendo ventajas de tipo organizativo gracias a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Además, la parte del servidor puede repartirse en varias máquinas y/o programas, lo que supone un aumento en el grado de distribución del sistema (Schiaffarino, 2019).

La función principal del cliente es mostrar visualmente un contenido, interpretando los archivos e información que ha recibido desde el servidor. El cliente, en el flujo web, corresponde al navegador que se utiliza para navegar por internet. Este software tiene la capacidad de interpretar varios lenguajes de programación especialmente creados para la visualización de contenidos. Por tanto, es el encargado de hacer acciones como construir los botones de la página, responder a los movimientos del usuario y adaptar textos y elementos gráficos según el espacio disponible en la pantalla.

Por otro lado, el servidor es el conjunto de hardware y software encargado de estar a la espera de las peticiones que le hace el cliente a través de internet. También de conectarse a otros servicios de internet, así como GCP. Habitualmente, el servidor actúa

de depósito y gestor de los datos que usa el cliente, interpreta ficheros y envía mensajes de respuesta de tamaño variable al cliente y/o a otros servicios de internet.

Algunos ejemplos de aplicaciones computacionales que usan este modelo son el correo electrónico y la World Wide Web. La aplicación presentada, que se encontrará en la web, utilizará también el modelo cliente-servidor.

## 4.3 Parte Cliente

Una vez es definido el concepto de cliente-servidor, es posible profundizar y definir los diferentes lenguajes de programación que son usados en el desarrollo del cliente de este proyecto, haciendo énfasis también en la librería ReactJS. Las tecnologías que se describen a continuación han sido seleccionadas para este proyecto ya que previamente, para las prácticas en empresa, he podido impartir un curso en el que he aprendido a desarrollar aplicaciones utilizándolas y me han ofrecido muy buenos resultados.

### 4.3.1 HTML

HTML, siglas que provienen de “HyperText Markup Language”, es el lenguaje de marcado más usado para la creación de páginas Web y sirve principalmente para indicar el orden del contenido. Este orden se indica usando marcas de hipertexto, también conocidas como tags. Además, los tags ayudan a los buscadores como Google, Yahoo o cualquier otro a encontrar la información, ya que el nombre del tag proporciona información sobre el contenido. Sin embargo, para este proyecto, al tratarse de una aplicación que será usada internamente por empresas, la visibilidad en los buscadores no tiene relevancia.

### 4.3.2 CSS

Las siglas CSS, que provienen de los términos ingleses Cascading Style Sheets, se trata de un lenguaje de diseño gráfico usado para determinar los estilos de cada uno de los elementos, tags, generados con el lenguaje HTML en una página web. Con el lenguaje CSS se puede conseguir el aspecto deseado de cualquier elemento de la interfaz. Por contra, aprender todas sus palabras clave y al utilizarlo profesionalmente puede llevar mucho tiempo de aprendizaje.

### 4.5.3 Javascript

Usado en un 95% de la páginas webs que hay a nuestra disposición hoy en día (Web Technology Surveys, 2019), Javascript es un lenguaje de programación orientado a objetos con el cual es posible realizar actividades complejas en una página web. Actualmente, ya existen muchas páginas de internet que no sólo muestran información estática, sino que también muestran datos actualizados a tiempo real, permiten interactuar con animaciones gráficas en 2D y 3D, y muchas otras características que permiten a los usuarios gozar de páginas dinámicas.

Además, debido al gran crecimiento, se han creado una gran cantidad de librerías muy útiles alrededor de este lenguaje. Para el desarrollo de la parte cliente de este proyecto, la librería que se usa es la conocida ReactJS, desarrollada por los ingenieros de Facebook. Más aún, también han surgido librerías con las cuales es posible desarrollar la parte de servidor con Javascript, como es el caso de NodeJS.

#### 4.3.4 ReactJS

ReactJS es una conocida librería de Javascript creada por Facebook en la cual es posible desarrollar el diseño de las pantallas que el cliente mostrará a los usuarios finales. La principal diferencia de ReactJS respecto a otras librerías es la organización y renderizado del código en componentes (F., 2017). Los componentes son piezas de código que contienen código en HTML (aunque también pueden contener otros componentes) que juntos forman los diferentes elementos que se pueden visualizar en la pantalla. Algunos de los beneficios del uso de componentes son:

- ❖ Renderizado de la página más óptimo. Cuando una aplicación de React está cargada en el navegador, si se realiza un cambio en un componente no se renderiza toda la página, sino que se renderiza solamente aquel componente que ha sufrido el cambio. Esto implica grandes mejoras de rendimiento en el navegador.
- ❖ Los componentes están formados por HTML, CSS y Javascript. De esta manera es mucho más fácil la integración de la lógica de la aplicación con los lenguajes de estilos.
- ❖ Cada componente es una clase que se puede instanciar, con la posibilidad de añadir los argumentos deseados, todas las veces que sean necesarias. De tal forma es muy sencillo reutilizar componentes en cualquier pantalla de la aplicación.
- ❖ Los componentes son asíncronos, lo que significa que la aplicación no cargará los componentes consecutivamente, sino que cada componente se carga independientemente de los demás, evitando así posibles bloqueos en la aplicación.
- ❖ Permiten el uso de librerías externas y los componentes de estas librerías. Además, existe una gran cantidad de librerías muy útiles y documentadas, como Material UI. Esta es una de las funcionalidades básicas por la que a ReactJS se le puede sacar tanto provecho.

#### 4.3.5 Librerías

En este proyecto, se han utilizado una gran cantidad de librerías para el cliente, tanto de Javascript como también algunas especializadas en ReactJS. A continuación se describen aquellas que han ganado más peso en el desarrollo de la aplicación.

#### 4.3.5.1 Material UI

Como se ha descrito en la sección 4.1.1, Material UI es una librería muy amplia que ofrece muchísimos componentes de ReactJS para adaptarlos a cualquier aplicación. Ofrece el diseño de las ventanas emergentes, botones, menús y pantallas de ejemplo, entre muchos otros recursos.

#### 4.3.5.2 MUI Datatable

Librería que ofrece un componente de ReactJS para crear tablas interactivas que usan los componentes y diseños de Material UI. Proporciona muchas opciones como por ejemplo filtros, redimensión de la tabla y exportación de datos. Este componente ha sido usado en las tablas que hay situadas en las pantallas de la aplicación.

#### 4.3.5.3 Axios

Librería de Javascript construida con el objetivo de gestionar la programación asíncronas de promesas HTTP a los servidores. Nos proporciona una función desde la cual es posible configurar, realizar y recibir respuestas fáciles de procesar (Caules, 2017). Con ella, desde la aplicación presentada se realizan todo tipo de peticiones al servidor encargado de conectarse a Firestore, y así, seguidamente, recibir los datos necesarios a mostrar en la interfaz.

#### 4.3.5.4 React Filepond

Librería de Javascript para ReactJS que ofrece un componente desde el cual es posible cargar archivos a la interfaz. Ofrece parámetros con los que configurar la subida de archivos, por ejemplo el tipo de archivo que se permite o la cantidad o tamaño máximo. También, está optimizado para que la carga en el navegador sea rápida.

#### 4.3.5.5 Redux

Librería que permite controlar y acceder al estado global de la aplicación entre los diferentes archivos que la conforman, de una forma consistente entre cliente y servidor. Esto sería análogo a gestionar variables globales, pero que continúan persistiendo entre los diferentes archivos. En la aplicación presentada es útil para mantener el objeto que representa al usuario en todas las pantallas. De tal forma, en una futura fase del proyecto se darán más o menos permisos según el rol de usuario que se tenga.

#### 4.3.5.6 Recharts

Recharts es una librería creada para ReactJS desde la cual es posible crear una gran variedad de gráficos y con muchas opciones para personalizarlos. Es utilizada en el gráfico que se muestra en la pantalla de Punto de venta.

#### 4.3.5.7 React Google Maps

Librería de Javascript para ReactJS que permite insertar, de forma relativamente sencilla, un mapa interactivo en la interfaz. Además, dispone de diversos parámetros configurables, así como la posibilidad de mostrar marcas en el mapa en las coordenadas deseadas.

### 4.4 Parte Servidor

Como se ha comentado, el servidor será el encargado de responder a las peticiones del cliente y de conectarse a los servicios de Google Cloud Platform. Al conectarse a los servicios de GPC es responsable de pedir la información a la base de datos de Google Cloud Firestore y procesarlos. También de subir imágenes a Google Cloud Storage y acceder a ellas para después ser mostradas en la interfaz.

La tecnología usada para el desarrollo del servidor de la aplicación es NodeJS. Node es un entorno de Javascript diseñado para ejecutarse en el lado servidor (Node.js). Fue generado para generar un sistema escalable y que tuviese consistencia como para poder generar un gran número de conexiones de formas simultánea. Actualmente, una de sus principales ventajas respecto a otros lenguajes para desarrollar el código de servidor es el gran rendimiento que ofrece.

Desde esta parte, los servicios de Google Cloud Platform que se usan o a los que se les requieren acciones son:

- ❖ APIs & Services
- ❖ Google App Engine
- ❖ Google Cloud Storage
- ❖ Google Cloud Firestore

### 4.5 APIs & Services

APIs & Services es un dominio de Google Cloud Platform desde el cual se gestiona al acceso a las APIs de Google, permite comunicación entre los servicios de Google y también la integración de estos en otros servicios. El uso de las APIs requiere autenticación y autorización por parte de la aplicación que quiere consumirlas, es por eso que la gestión de las credenciales es básica para poder completar el proyecto.

Existen tres tipos de credenciales: API keys, OAuth 2.0 client IDs y Service account keys. En el desarrollo del sistema que cubre en este trabajo son necesarias las tres credenciales. A continuación se detalla cada una de ellas:

- ❖ API keys: clave encriptada que identifica al proyecto de Google para poder hacer un cobro por uso. Las API keys se utilizan para llamar a las APIs de Google que no requieren autenticación, así como la API de Google Maps que permite mostrar el mapa en la interfaz de la aplicación e interactuar con él y la API Geocoding, que transforma una dirección escrita a coordenadas geográficas. Ambas son usadas en este proyecto.
- ❖ OAuth 2.0 client IDs: sistema para identificar usuarios. Google utiliza la autorización OAuth 2.0 para gestionar el registro de los usuarios en la aplicación. Estas credenciales son aplicadas en la pantalla de Iniciar sesión para que, una vez el usuario haga clic, sea redirigido a la página de autenticación de Google, desde donde podrá utilizar una cuenta de la compañía para entrar o registrarse en la plataforma.
- ❖ Service accounts: clave que representa la aplicación. Se proporciona en el código de la aplicación y se usa para poder acceder a algunos servicios de Google Cloud Platform, así como Google Cloud Storage y Google Cloud Firestore.

## 4.6 Google App Engine

Google App Engine se define como una plataforma de aplicaciones sin servidor totalmente gestionada. En esta es posible crear y desplegar aplicaciones sin la necesidad de preocuparse por la infraestructura requerida al publicar una aplicación y las complicaciones a las que esto puede conducir (Google Cloud, 2019). Además, App Engine cuenta con un balanceador de carga y escalamiento automático, cosa que hace que no nos tengamos que preocupar por los servidores físicos que están abiertos sirviendo la aplicación. Google se encarga de encender las máquinas necesarias según el tráfico para que la respuesta del servicio sea óptima, dentro de la capacidad contratada. De tal modo, los desarrolladores pueden centrarse 100% en la lógica y desarrollo sin la necesidad de crear una sobrecarga de esfuerzo desplegando la infraestructura.

Este servicio de google permite el despliegue de aplicaciones con el servidor escrito en NodeJS. Por tanto, la plataforma desarrollada en este proyecto puede ser apoyada en App Engine.

En el momento de desplegar una aplicación a App Engine, es importante seleccionar el tipo de máquina más adecuado para la aplicación según la memoria que necesite, la frecuencia máxima de la CPU y si queremos que se escalen o no automáticamente. Cuantos más recursos, más ascenderá su coste. Los diferentes tipos de máquina a seleccionar son:

| Clase            | Límite de memoria | Límite de CPU | Escalamiento |
|------------------|-------------------|---------------|--------------|
| F1 (por defecto) | 128 MB            | 600 MHz       | automático   |
| F2               | 256 MB            | 1,2 GHz       | automático   |
| F4               | 512 MB            | 2,4 GHz       | automático   |
| F4_1G            | 1024 MB           | 2,4 GHz       | automático   |
| B1               | 128 MB            | 600 MHz       | manual       |
| B2               | 256 MB            | 1,2 GHz       | manual       |
| B4               | 512 MB            | 2,4 GHz       | manual       |
| B4_1G            | 1024 MB           | 2,4 GHz       | manual       |
| B8               | 1024 MB           | 4,8 GHz       | manual       |

*Tabla 4.1: Clases de máquinas disponibles en Google App Engine.*

Para esta primera fase del proyecto y tras pruebas con los varios tipos de máquinas, se decide optar por la clase F4, la cual ofrece un rendimiento suficientemente correcto.

## 4.7 Cloud Firestore

El servicio de Google Cloud Firestore, descrito en la sección 2.2 del trabajo, es sumamente usado por la aplicación web. Constantemente, el servidor de la aplicación realiza peticiones y publica campos en la base de datos de Firestore. A continuación se describen todos los casos en que se requiere la participación de Firestore:

- ❖ Autenticación de usuarios.
- ❖ Obtener la lista de usuarios.
- ❖ Dar acceso a los usuarios.
- ❖ Conseguir la lista de visitas.
- ❖ Conseguir la lista de las visitas para un único punto de venta.
- ❖ Obtener todos los datos de una única visita.
- ❖ Publicar los datos de nuevas visitas.
- ❖ Eliminar visitas.
- ❖ Conseguir la URL de las imágenes de lineales que hay guardadas en Storage.
- ❖ Conseguir la lista de puntos de venta.
- ❖ Conseguir los datos de un punto de venta.
- ❖ Asignar las coordenadas del mapa a un punto de venta.

- ❖ Crear nuevos puntos de venta.
- ❖ Guardar el cambio de los facings acordados en un punto de venta.

## 4.8 Cloud Storage

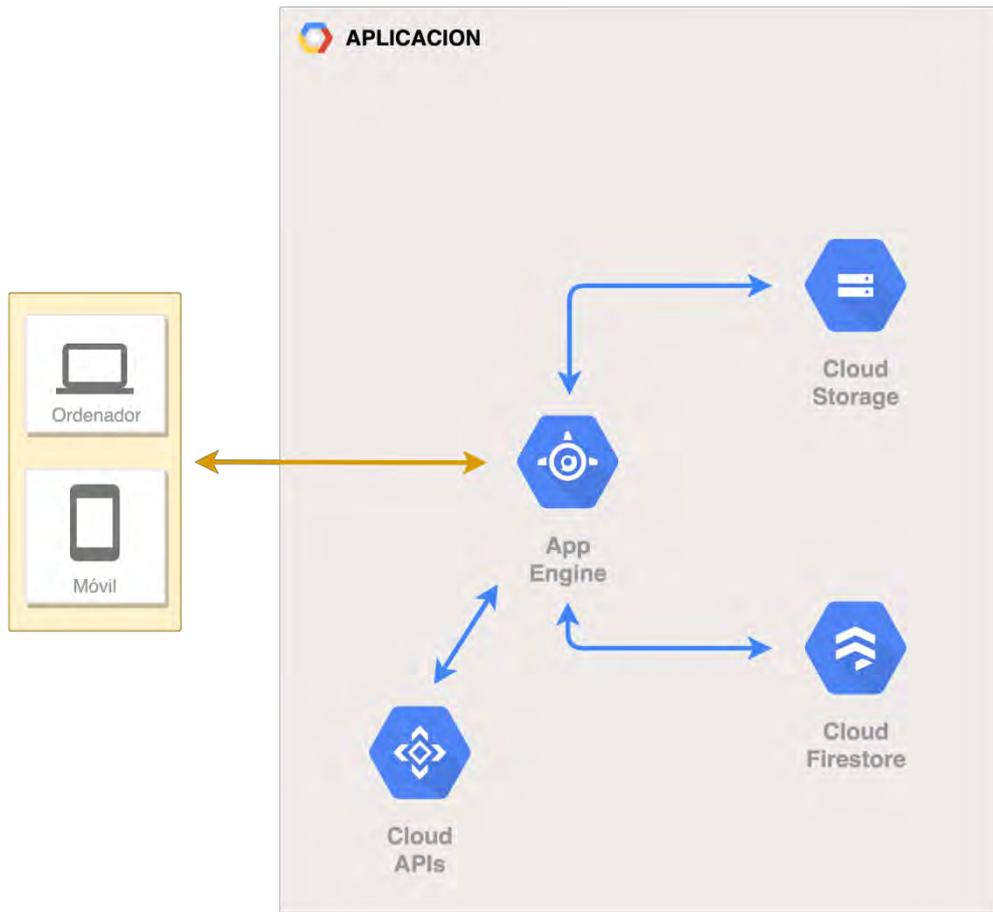
Storage, el servicio de almacenamiento de archivos de Google Cloud Platform descrito en la sección 2.1, también es frecuentemente utilizado por la aplicación para leer, publicar y leer archivos. La aplicación utiliza Cloud Storage para:

- ❖ Obtener los resultados de la predicción del modelo, los cuales están almacenados en un archivo JSON.
- ❖ Publicar las imágenes de lineales o expositores que se suben a la interfaz.
- ❖ Eliminar las imágenes de aquellas visitas que son eliminadas.

## 4.9 Diagrama de la arquitectura de la aplicación

Conociendo todos los elementos que hacen posible el funcionamiento y gestión de toda la aplicación, se presenta en la Figura 2.24 la arquitectura del sistema que la conforma.

En el diagrama se puede observar la conexión entre los dispositivos finales (navegadores de ordenadores preferiblemente, pero también de móviles) y App Engine, el cual proporcionará el acceso a la aplicación y a todos sus servicios. App Engine por otro lado, es el servicio encargado de conectarse a las APIs de Google, a Cloud Storage y a Cloud Firestore por separado.



*Figura 4.24: Diagrama de la arquitectura de la aplicación.*

## 5. CONCLUSIONES

### 5.1 ¿Qué se ha conseguido?

Una vez analizado la aplicación resultante del proyecto y la precisión de predicción del modelo de aprendizaje automático, se puede decir que se han alcanzado los objetivos que se planteaban en un primer momento: desarrollar un sistema completo capaz de realizar la verificación de cumplimiento de planogramas. Para ello se han utilizado las últimas tecnologías y técnicas disponibles, así como la computación en la nube junto a la tecnología web.

En cuanto al conjunto de sistemas que hacen posible la detección y clasificación de productos en imágenes, se ha conseguido una arquitectura robusta en la nube que ofrece un buen rendimiento en todo el proceso, desde que se recibe una imagen hasta que se obtienen los resultados con la detección de los productos. Esto ha sido posible gracias al uso de los servicios de Google Cloud Platform, desde donde se ofrecen soluciones a la mayoría de situaciones con las que se puede encontrar un desarrollador.

El hecho de utilizar AutoML Vision Object Detection suponía un reto ya que es un servicio de Google muy reciente por lo que no se tenían referencias sobre su eficacia. Además, al estar aún en fase beta, implica que se puedan producir cambios en el servicio que provoquen incompatibilidad con versiones anteriores o pequeños errores aún por resolver. Sin embargo, los resultados obtenidos de AutoML han sido positivos, ofreciendo una muy buena detección y clasificación de los productos.

Por otro lado, la aplicación web resultante que da acceso a todo el sistema es perfectamente funcional y accesible desde cualquier navegador. En ella se puede auditar el cumplimiento de planogramas de todas las visitas que se realizan en los diferentes puntos de venta. Adicionalmente, está desplegada en la infraestructura de App Engine desde donde ofrece un buen rendimiento.

A nivel personal, el desarrollo de este proyecto me ha dado la capacidad de comprender y establecer las relaciones adecuadas que se fijan entre los diferentes productos de una plataforma servicios web. El hecho de aprender nuevos lenguajes de programación así como Javascript y el uso de todas sus librerías también me ha abierto el área de conocimiento que tenía hasta la fecha del comienzo del proyecto. Por tanto, el conjunto del trabajo, todas las fases que se han pasado para hacerlo posible, las valoro como una experiencia muy positiva.

## 5.2 Trabajo futuro

Tal y como se ha comentado en un primer momento, el proyecto se trata de una primera fase, y es que en un proyecto con tanto recorrido siempre quedan tareas o funcionalidades por añadir para mejorarlo y darle más valor, más allá de conseguir el objetivo base. A continuación se detallan futuras tareas a realizar.

- ❑ Comprobar la usabilidad del producto. Para ello sería necesario realizar un test de usabilidad a distintos usuarios para sacar conclusiones sobre la interfaz.
- ❑ Entrenar un modelo de AutoML Vision Object Detection para otros tipos de productos. Más aún, se podría entrenar para detectar los productos de la competencia y extraer datos útiles.
- ❑ Entrenar modelos que sean capaces de detectar precios y ofertas.
- ❑ Ofrecer a los GPVs la posibilidad de consultar la ruta de los puntos de venta que han de visitar.
- ❑ Añadir una página desde la cual gestionar los productos disponibles de la marca, para poder añadirlos y eliminarlos.
- ❑ Detectar dos imágenes iguales cargadas a la aplicación. De este modo se podría evitar el fraude de crear una visita que realmente no se ha hecho.

## Bibliografía

Bisong, E. (10 de Mayo de 2019). *Medium*. Obtenido de Google Cloud AutoML Vision for Medical Image Classification: <https://towardsdatascience.com/google-cloud-automl-vision-for-medical-image-classification-76dfbf12a77e>

Boubeta, A. I. (2006). *Organización en el punto de venta. Manual Básico de Gestión de Tienda*. Vigo: Ideaspropias Editorial.

Caules, C. Á. (21 de Diciembre de 2017). *Arquitectura java*. Obtenido de Blog sobre Java EE - Axios js una librería de promesas: <https://www.arquitecturajava.com/axios-js-una-libreria-de-promesas/>

Cloud, Google. (s.f.). *Google Cloud*. Obtenido de Ubicaciones de GCP: <https://cloud.google.com/about/locations/>

Cloud, Google. (8 de Junio de 2019). *Google Cloud*. Obtenido de AI & Machine Learning Products: <https://cloud.google.com/vision/automl/object-detection/docs/prepare>

Cloud, Google. (29 de Abril de 2019). *Google Cloud*. Obtenido de Descripción general de la autenticación: [https://cloud.google.com/docs/authentication/?hl=en\\_GB&\\_ga=2.153094264.-1788957440.1550488052](https://cloud.google.com/docs/authentication/?hl=en_GB&_ga=2.153094264.-1788957440.1550488052)

Datakey, I. (21 de Mayo de 2014). *Datakey investigación de mercados y sondeos de opinión*. Obtenido de Merchandising: Definición y función del lineal: <http://www.instituto-datakey.com/1025/>

F., J. (2 de Julio de 2017). *Codeburst.io*. Obtenido de What is ReactJS and how can it make your life easier: <https://codeburst.io/what-is-reactjs-and-how-can-make-your-life-easier-3beb797f30e4>

Google Cloud. (s.f.). *Google Cloud*. Obtenido de Google Cloud Functions: <https://cloud.google.com/functions/>

Google Cloud. (6 de Junio de 2019). *Google Cloud*. Obtenido de The App Engine Standard Environment : <https://cloud.google.com/appengine/docs/standard/>

InboundCycle, Equipo. (22 de Octubre de 2016). *Inboundcycle*. Obtenido de Blog de Inbound Marketing - ¿Qué es el trade marketing?: <https://www.inboundcycle.com/blog-de-inbound-marketing/que-es-el-trade-marketing>

Kluwer, Wolters. (s.f.). *Wolters Kluwer*. Obtenido de Diccionario empresarial: [http://diccionarioempresarial.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAEAMtMSBFljTAAASNjc3MLtbLUouLM\\_DxbIwMDS0NDA1OQQGZapUt-ckhlQaptWmJOcSoARLj7xTUAAAA=WKE](http://diccionarioempresarial.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAEAMtMSBFljTAAASNjc3MLtbLUouLM_DxbIwMDS0NDA1OQQGZapUt-ckhlQaptWmJOcSoARLj7xTUAAAA=WKE)

Loop, H. i. (30 de Octubre de 2018). *Hackernoon*. Obtenido de The best image annotation platforms for computer vision (+ an honest review of each): <https://hackernoon.com/the-best-image-annotation-platforms-for-computer-vision-an-honest-review-of-each-dac7f565fea>

Medium. (25 de Abril de 2018). *Medium*. Obtenido de Google Cloud Functions Tutorial : What is Google Cloud Functions?: <https://medium.com/@iromin/google-cloud-functions-tutorial-what-is-google-cloud-functions-8796fa07fc7a>

Medium. (17 de Septiembre de 2018). *Medium*. Obtenido de AutoML Developments and Future: [https://medium.com/@Packt\\_Pub/automl-developments-and-future-788a710237c](https://medium.com/@Packt_Pub/automl-developments-and-future-788a710237c)

Metafase. (s.f.). *Metafase, profesionales de la venta externa*. Obtenido de Gestor Punto de Venta (GPV): <https://www.metafase.com/gestor-punto-de-venta-gpv/>

Node.js. (s.f.). *Node.js*. Obtenido de About Node.js®: <https://nodejs.org/en/about/>

Sanchom. (1 de Septiembre de 2011). *Sanchom*. Obtenido de Tag: average precision: <https://sanchom.wordpress.com/tag/average-precision/>

Schiaffarino, A. (12 de Marzo de 2019). *Infranetworking*. Obtenido de Modelo cliente servidor: <https://blog.infranetworking.com/modelo-cliente-servidor/>

Sonneveld, K. (2000). What drives (food) packaging innovation? . *Packaging Technology and Science* , 29-35.

Web Technology Surveys. (1 de Mayo de 2019). *Web Technology Surveys*. Obtenido de W3Techs - World Wide Web Technology Surveys: <https://w3techs.com/>

Xplenty. (28 de Septiembre de 2017). *Xplenty Blog*. Obtenido de The SQL vs NoSQL Difference: MySQL vs MongoDB: <https://medium.com/xplenty-blog/the-sql-vs-nosql-difference-mysql-vs-mongodb-32c9980e67b2>

## Anexo 1

Para entender la evaluación de los modelos de inteligencia artificial, es importante conocer las métricas usadas. En este anexo se detallan los conceptos de IoU, precisión y sensibilidad (“recall”).

- ❑ Precisión: corresponde a la frecuencia de predicciones que fueron correctas, positivas. Una precisión más alta implica una menor cantidad de falsos positivos predichos. Este valor se calcula para un threshold de confianza concreto. La fórmula usada para el cálculo es la siguiente:

$$\text{precisión} = \frac{\text{verdaderos positivos}}{\text{verdaderos positivos} + \text{falsos positivos}}$$

- ❑ Sensibilidad (o “Recall”): corresponde a la frecuencia de *bounding boxes* que fueron predichas correctamente por el modelo. Una sensibilidad más alta implica menos falsos negativos, que también se puede entender como que se omitirán menos predicciones. Este valor se calcula para un threshold de confianza concreto. La fórmula usada para el cálculo es la siguiente:

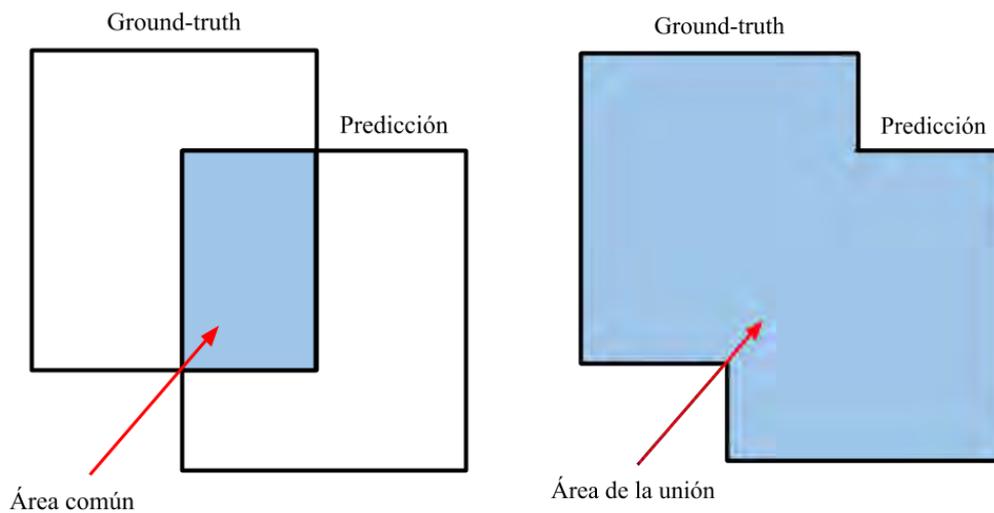
$$\text{sensibilidad} = \frac{\text{verdaderos positivos}}{\text{verdaderos positivos} + \text{falsos negativos}}$$

- ❑ Precisión media o Area under Precision and Recall Curve (AuPC): valor calculado realizando la suma del valor de las precisiones obtenidas para todos los valores de threshold posibles multiplicado por el incremento en la sensibilidad (Sanchom, 2011). La fórmula correspondiente es la siguiente:

$$\text{AuPC} = \sum_{k=1}^N P(k) \cdot \Delta r(k)$$

Donde  $N$  es el nombre total de imágenes en la colección con a que se evalúa el sistema,  $P(k)$  es la precisión en un límite de  $k$  imágenes y  $\Delta r(k)$  es el incremento en la sensibilidad entre  $k$  y  $k - 1$  imágenes.

- ❑ IoU: sigla de “Intersection over Union”, es una métrica de evaluación común usada para medir la exactitud en los sistemas de detección de objetos. Para calcular el valor es necesaria tanto la *bounding box* del etiquetado (ground-truth) como la *bounding box* que el modelo ha predicho. El valor corresponde a la división del área común de ambas *bounding boxes* entre el área que forma la unión de ambas. Podemos ver esta idea ilustrada en la Figura X.



**Figura A.1:** De izquierda a derecha, representación del área común entre dos bounding boxes y el área de la unión.

De tal modo, el valor IoU se calcula con la fórmula:

$$IoU = \frac{\text{Área común}}{\text{Área de la unión}}$$