



Tu mejor bar en un plis plas

MY BAR

Proyecto Big Data I

31 enero de 2022

Aplicación de búsqueda de bares en tiempo real

VICTOR SALVADOR LOPEZ
victorsalvadorlopezz@gmail.com

Contenido

Resumen.....	2
Introducción y justificación teórica	2
Motivación	2
Beneficios	2
Estado del arte y competidores	3
Objetivos	7
Objetivo principal:	7
Objetivos secundarios	8
Solución propuesta	8
Tecnologías Intervinientes	9
Plan de trabajo	9
Infraestructura	9
Tratamiento de datos.....	10
Modelado	10
Metodología.....	10
Fase 1:	11
Fase 2:	11
Presupuesto	11

Resumen

My Bar es una aplicación para IOS y Android donde puedes encontrar en tiempo real bares con aforo disponible mediante tu localización, de esta manera se facilita las salidas por ocio agilizando el proceso de búsqueda del local. También dispone de un sistema de reservas para aquellos locales inscritos a nuestra aplicación mediante una serie de pasos a seguir. Esto permite a los comercios impulsar el número de clientes y optimizar sus reservas. Para aquellos locales donde no se puedan obtener datos fiables sobre el aforo, se rastreará la afluencia en la zona y se realizará una estimación recogiendo datos de distintas fuentes. Toda esta información se actualiza en tiempo real por lo que se requiere de una infraestructura de tecnología Big Data para sostener todas las transacciones de la aplicación.

Introducción y justificación teórica

¿Cuántas veces nos hemos desesperado buscando un sitio donde tomar algo un viernes por la noche, y hemos terminado en un bar cutre que ni siquiera nos gustaba? La nueva era de la información nos permite utilizar herramientas para publicar todo tipo de contenido en tiempo real y retransmitirlo a todas las partes del mundo; es una realidad que las emergentes tecnologías “in real-time” y su integración con la sociedad, están modificando la vida cotidiana de las personas. Un ejemplo de éxito es la aplicación Waze, donde se procesan todo tipo de datos de tráfico en tiempo real (como afluencias, accidentes, obras, controles policiales y radares de velocidad, vehículos detenidos, avisos, etc.) de millones de solicitudes al segundo por parte de sus usuarios, y hoy en día ya es imprescindible para algunos conductores.

La vocación de MyBar es eliminar, o como mínimo agilizar, este tipo de situaciones. Mediante el uso de herramientas Big Data y de Inteligencia Artificial, será capaz de estimar los peatones en una determinada zona de la ciudad a una concreta hora del día; con la intención de informar al consumidor y sugerir la conveniencia de salir en una zona u otra.

Motivación

Tras una breve experiencia ociosa en las calles de Madrid uno se percató de la cantidad de bares y locales que hay donde salir, pero también de las conmensuradas cantidades de gente que acude a ellos; por lo que a veces resulta realmente molesto ir buscando un local y consecutivamente encontrar aforos llenos o sitios poco recomendables. De aquí nace la motivación de MyBar, de ofrecer una herramienta a sus usuarios (público joven) para que este tipo de situaciones terminen de una vez. Nos motiva porque es un vacío que nadie ha llenado correctamente hasta la fecha, y porque este segmento de la demanda se caracteriza por enormes impulsos de popularidad a aplicaciones que realmente les aportan algo nuevo, útil y relacionado con su ocio.

Beneficios

- Poder encontrar bares abiertos en tiempo real
- Poder averiguar valoraciones de bares cercanos al instante
- Poder estimar el aforo en un bar en función de la hora y la zona
- Poder visualizar el local a través de nuestras imágenes

Estado del arte y competidores

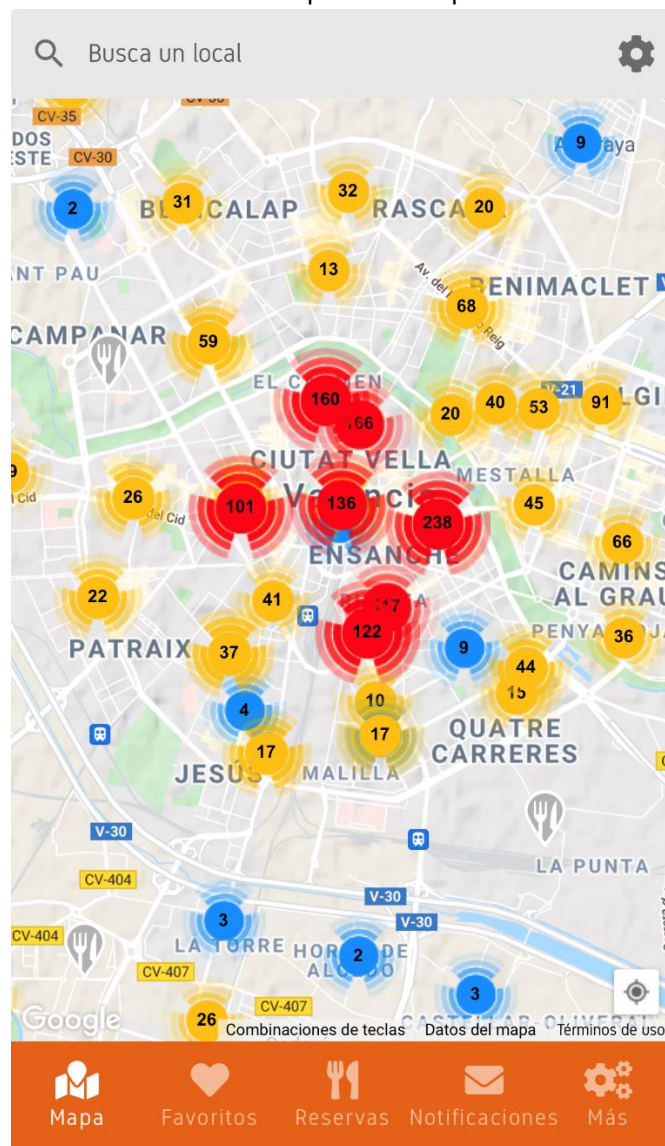
Existen dos competidores a MyBar en España:

- Findyt (<https://findyt.es/>):
Se trata de una empresa valenciana que lanzó su aplicación en 2020. En su página describen las siguientes funcionalidades:

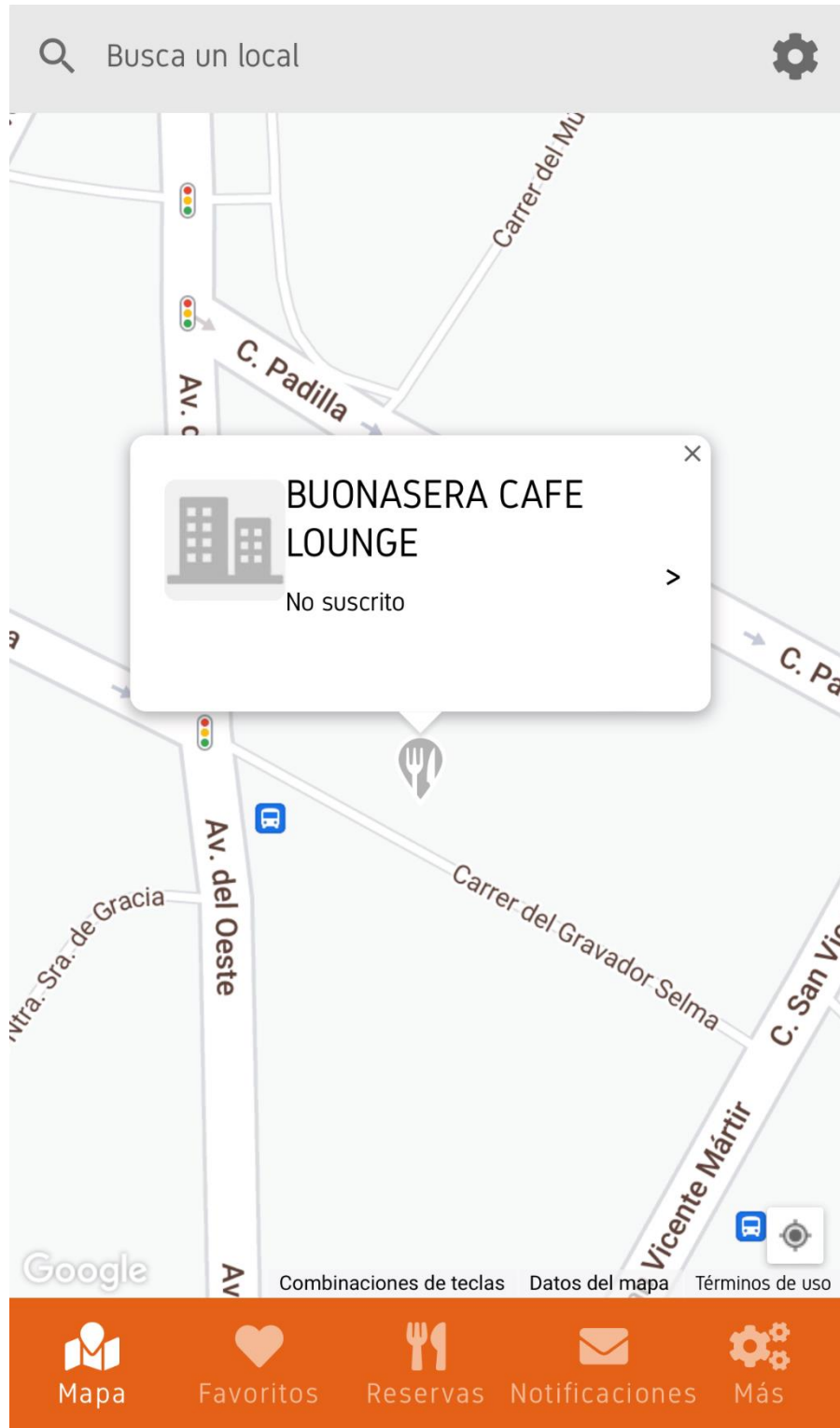


Hemos comprobado sus funcionalidades en la aplicación para IOS:

Mostramos la ciudad de Valencia porque es de donde proviene la compañía. Se indican zonas pobadas de bares con colores a especie de mapa de calor.



A continuación, seleccionamos un local. No se ha encontrado ningún solo bar o restaurante que estuviese suscrito a la aplicación, por lo que vemos que esta app está en desuso



Si accedemos al local solamente obtenemos una imagen predeterminada de un bar que aparece en todos los bares seleccionados, su dirección, número de teléfono y enlace a la web.



Grabador Selma, 4 LOCAL, 46001, València



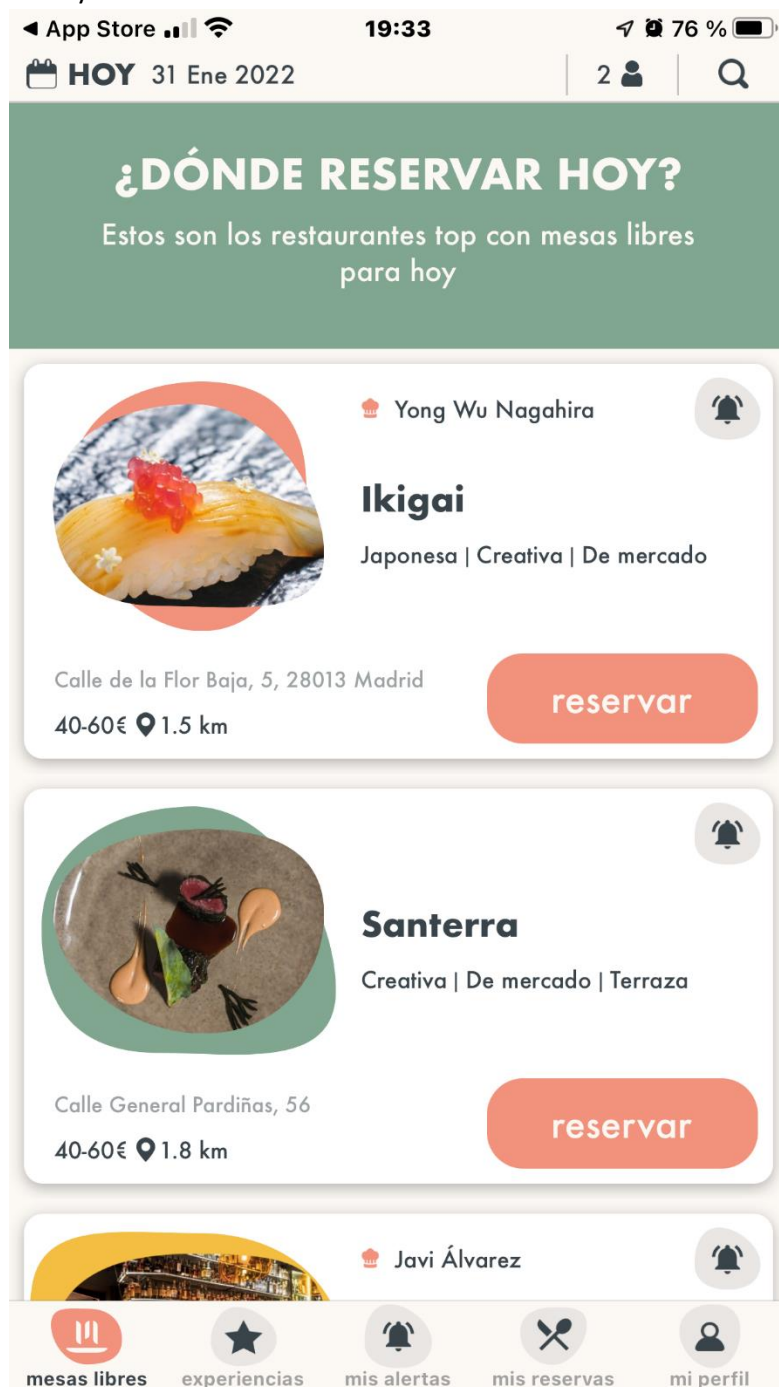
En conclusión, esta aplicación no cumple las promesas de disponibilidad en tiempo real, solamente se limita a imitar a Google Maps, posiblemente mediante su API y ofrecer un servicio más reducido que la misma y casi con ninguna mejora.

- Maybein (<https://www.maybein.com/>):

Según la revista EL ECONOMISTA, “Maybein, que nació en junio de año pasado en Valladolid, es fruto de una experiencia personal de su fundadora y consejera delegada, Saray Maestro, sorprendida al comprobar que había mesas libres en restaurantes cuya reserva a ella le aplazaban días o semanas.”

Básicamente se trata de una aplicación que tienen un convenio con distintos restaurantes de buen nivel, donde se gestionan las reservas de mesa. Puedes reservar una mesa o esperar a que se cancele una mediante la habilitación de notificaciones. También existe otra aplicación llamada Maybein Chef, cuyo servicio se vuelca hacia la otra orientación: permite a los restaurantes gestionar sus reservas con la interacción de las dos aplicaciones en el mismo backend.

Así se ve la interfaz principal de búsqueda de reservas en la aplicación, te permite activar notificaciones, cambiar fechas, reservar y buscar restaurantes que incluya la aplicación y modificar el número de comensales. La aplicación estima la distancia al establecimiento y el coste estimado de la comida.



Una vez seleccionado un restaurante, ofrece información relevante acerca del local, incluyendo la carta y una descripción del lugar



A diferencia de la primera esta sí que es una aplicación útil y competitiva de la que sus clientes pueden sacar provecho. La interfaz es intuitiva y el backend está ingeniosamente diseñado. Además de una más que respetable red de restaurantes añadidos. Por suerte, la actividad de la aplicación difiere de la de Mybar, debido a que ellos se enfocan en reservas de restaurantes y nosotros en bares disponibles en tiempo real. Una es para planificar una cita o una comida y la otra para encontrar un sitio rápidamente donde salir con amigos.

Objetivos

Objetivo principal:

El objetivo general es crear una aplicación accesible desde todo tipo de dispositivo telefónico moderno con red, capaz de recolectar datos mediante la herramienta Google Maps API de los

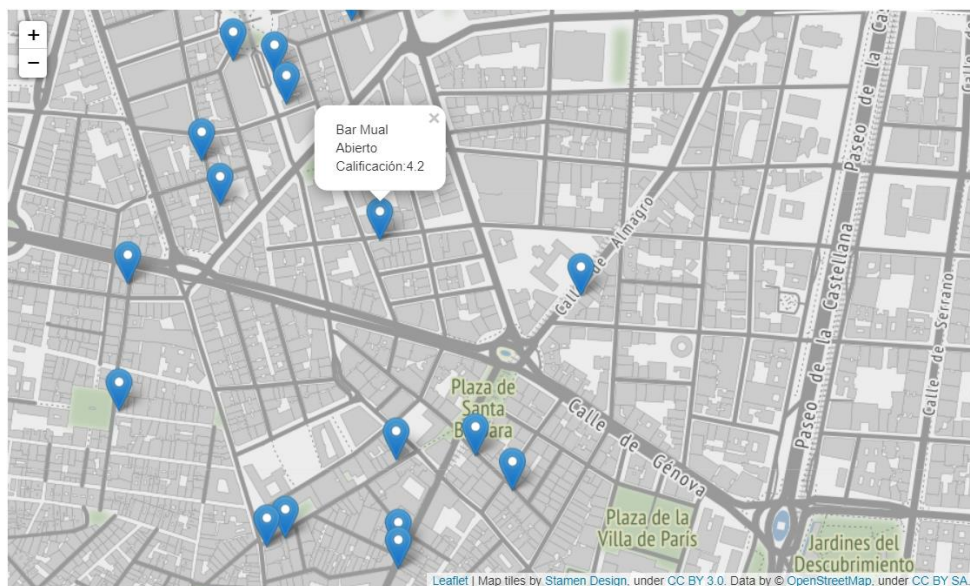
bares más cercanos al usuario; y con ellos diseñar un modelo predictivo con datos históricos para estimar el número de peatones en esa zona para ese determinado momento:

Objetivos secundarios

- Establecer un proceso de verificación de los usuarios
- Realizar hosting de la aplicación en alguna plataforma (firebase)
- Utilizar github para el control de versiones del proyecto
- Crear una página web donde introducir las características de la aplicación
- Interfaz de la aplicación para IOS y Android
- Introducir anuncios de Google para rentabilizar el coste de las solicitudes a la API en versiones más avanzadas
- Trabajar con gestiones de reservas en versiones más avanzadas

Solución propuesta

Se propone el desarrollo de un sistema que tiene 3 procesos principales: la primera es la solicitud a la API consultando bares en tiempo real dada una petición por parte del usuario, se le mostrará un mapa con los resultados y este podrá acceder a información relevante como su valoración, imágenes etc.



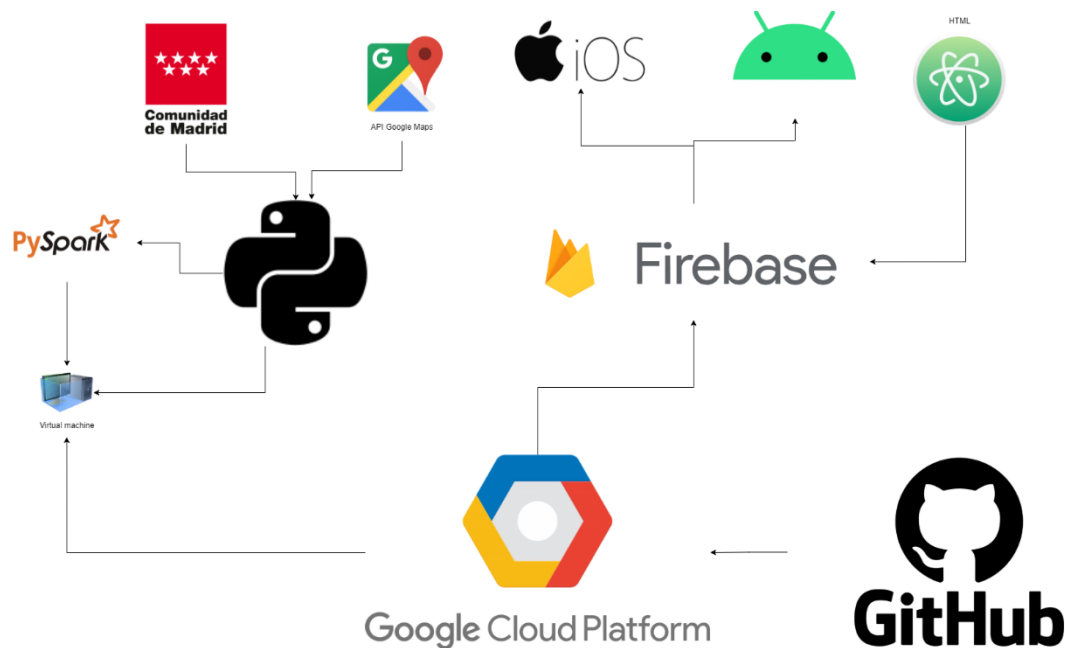
El segundo proceso interviniente es el desarrollo de un modelo predictivo del aforo de las calles utilizando registros históricos de la siguiente fuente: [Registro histórico Madrid](#). En primer lugar, se utilizarán datos del último año, aunque en futuras versiones se planteará entrenar el modelo con un conjunto mayor de datos utilizando Spark en una VM alquilada en GCP a través de un Google Colab para entrenar los datos históricos de otros años. Se probarán distintos modelos con RapidMiner y se modelará en Python para convenir la compatibilidad de las solicitudes de la Api y los datos registrados.



Por último, la solución pasa por el último proceso, que es aplicar los resultados de la búsqueda de la API e introducirlos en el modelo para que devuelva estimaciones del aforo. Esta técnica se deberá de ir perfeccionando conforme el paso de las versiones, así como el rendimiento del modelo y la compatibilidad de las dos fuentes de datos. También requerirá adaptaciones de los resultados para estimar de la mejor manera posible la disponibilidad de los locales encontrados.

Tecnologías Intervinientes

- Anaconda: Python para el tratamiento de datos
- Apache Spark: entrenamiento de modelos de inteligencia artificial en versiones avanzadas el proyecto
- Atom: edición de HTML
- GCP Environment:
 - Google Maps API: para la consulta de bares en tiempo real
 - VM: para aprovechar los procesos en Spark en procesamiento distribuido
 - Docker: virtualización
 - Firebase: control del proyecto
 - Servidor web
 - Almacenamiento
- Github: control de versiones del proyecto
- Rapidminer: pruebas de modelos



Plan de trabajo

Infraestructura

Para la elaboración de la aplicación, el primer paso será disponer de las infraestructuras necesarias para procesar multitud de peticiones en cortos de tiempo. Esta decisión debe adaptarse a la situación de uso de la aplicación. En el lanzamiento de la app simplemente se

producirán instancias en Cloud Run, con la intención de no alquilar una máquina virtual con sus correspondientes costes de mantenimiento para tareas que no requieran altos niveles de procesamiento. Una vez crezca la aplicación en cuanto a popularidad sí que deberemos virtualizar nuestra aplicación en un Docker y utilizar herramientas como App Engine, añadiendo un nivel de abstracción a la aplicación y optimizando el coste de las consultas. El modelo predictivo del aforo deberá de ejecutarse una sola vez alojado en GCP, puesto que no sería eficiente en coste y tiempo entrenar el modelo cada vez que se ejecutase una búsqueda.

Como ya se mencionó, el frontend de la aplicación estará alojado a través del framework Firebase, por lo que obtendremos analíticas sobre el lanzamiento de nuestra aplicación y podremos gestionar las autenticaciones de los usuarios. No se requerirá almacenamiento de momento, a no ser que queramos tratar en versiones futuras con imágenes de consultas ya almacenadas para agilizar el proceso de búsqueda de bares.

Tratamiento de datos

No se requiere un estudio exploratorio de los datos de búsqueda sino de un proceso lo más eficiente posible donde limpiemos y estructuremos los datos de las consultas API a Google Maps Platform. Preparando así los datos para poder ser introducidos en el modelo predictivo del aforo

Modelado

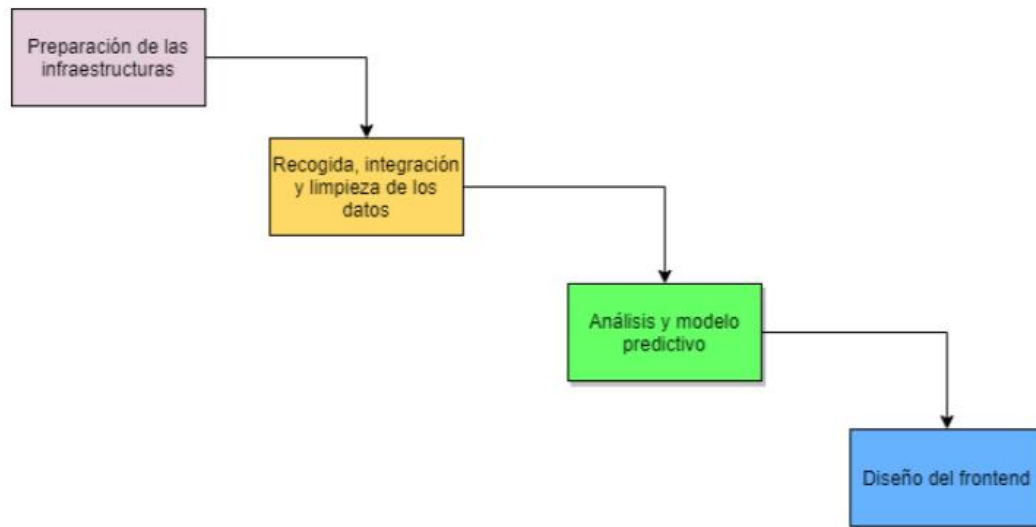
El primer modelo solo incluirá los datos del último ejercicio, será un proceso preparado para integrarse con los datos de las búsquedas y sí que requerirá de un análisis exploratorio y un extenso preprocesado de los datos para optimizar el rendimiento del modelo. Será elaborado en Python.

El segundo modelo incluirá todos los años de registro de aforo en las calles de Madrid en función de fechas y horas que se incluyan en los datos abiertos de la comunidad. Este modelo también requerirá de una capacidad procesamiento mayor, por lo que emplearemos Spark, en concreto la extensión Pyspark para facilitar la integración del modelo con el resto del proyecto. A su vez se requerirá de un clúster para aprovechar el procesamiento distribuido de Spark.

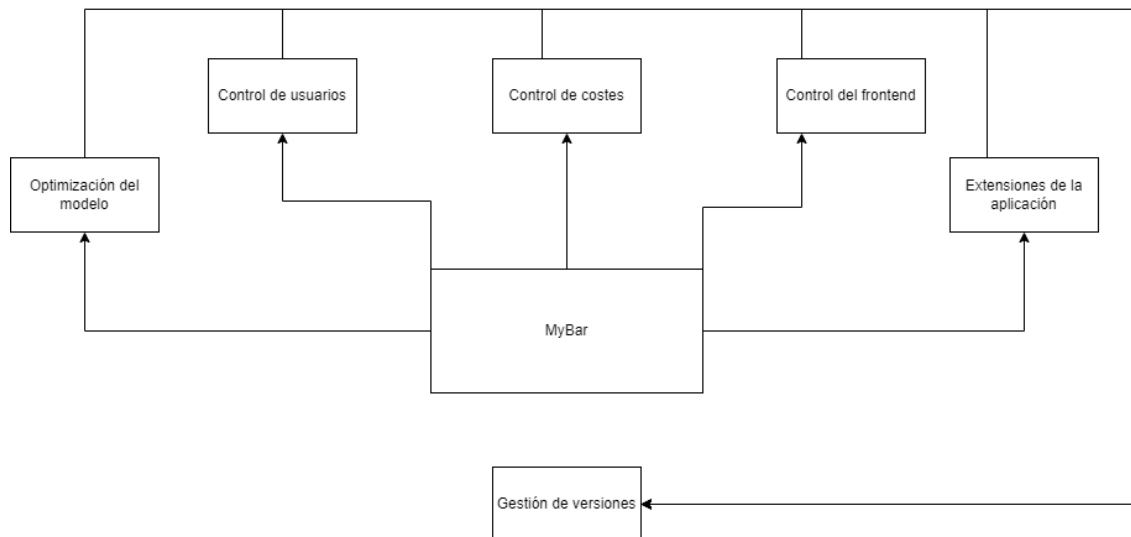
Metodología

Según la solución propuesta, nuestro proyecto consta de 4 fases: preparación de las infraestructuras, obtención y preprocesamiento de los datos, minería y modelaje, y creación del frontend. A estas deberíamos de añadirle una segunda etapa postpublicación, que implicaría tareas como el seguimiento de los datos, la revisión de los modelos de predicción, el diagnóstico de los costes e ingresos del uso de la aplicación, etc. Por ende, constamos de dos fases: una primera que implementará en un modelo de cascada, donde ejecutaremos paso por paso cada una de las partes descritas; y una segunda que seguirá un modelo ágil como SCRUM, donde una vez esté disponible nuestro servicio, este seguirá incrementando su cantidad de información, realizando predicciones y optimizando sus modelos; siempre atendiendo a posibles modificaciones internas a realizar en nuestro proyecto.

Fase 1:



Fase 2:



Presupuesto

Durante el lanzamiento de la aplicación los costes estimados son 0, debido a que solo se requerirán pruebas gratuitas para implementar la aplicación.

En el futuro sí se necesitarán utilizar herramientas con coste pero su estimación en coste es ambigua puesto que depende de la cantidad de solicitudes que se manejen de las distintas herramientas y del uso que finalmente se les de a cada una de ellas.