



Iñigo Martínez López

Premios a las mejores reutilizaciones de datos abiertos de Euskadi 2018

1. Introducción

Actualmente, los usuarios de dBizi puede conocer en tiempo real, con una frecuencia de 5-6 minutos, la disponibilidad de bicicletas en las 16 estaciones de la ciudad. Esta información no le garantiza al usuario disponer de bicicletas en su estación de salida, dada la afluencia de usuarios en determinadas horas. De igual modo, es posible que el usuario se encuentre sin bases disponibles en las que estacionar su bicicleta.

dBizi++ se presenta como una aplicación para la consulta y planificación de viajes en bicicleta con dBizi. Este servicio emplea datos de bicicletas disponibles con el objetivo de garantizar la disponibilidad del servicio en todo momento, y para todos los usuarios. La aplicación informa al usuario de la viabilidad de su trayecto, empleando datos históricos y en tiempo real.

El proyecto está destinado tanto a los usuarios de dBizi como al responsable del mantenimiento del sistema de bicicletas. Esta extensión del servicio dBizi permite a los usuarios estimar si dispondrán de bicicletas cuando acudan a su estación de salida y si, dada una estación de destino, tendrá bases libres para estacionar su bicicleta.

Asimismo, la información introducida por el usuario en la aplicación permitirá al responsable del mantenimiento conocer con antelación los posibles viajes de los usuarios y planificar acciones para garantizar la disponibilidad de bicicletas y bases libres.

2. Objetivos

- Para los usuarios: Mejorar la disponibilidad del servicio de dBizi
- Para el mantenedor del sistema: Conocer las estaciones más empleadas, los trayectos más habituales entre usuarios y los periodos de mayor uso del servicio.

3. Ventajas

- Consultar la viabilidad de un trayecto en tiempo real.
- Posibilidad de reservar, bajo coste, un trayecto concreto: estación origen, estación destino, y hora estimada.
- Visualización de los trayectos realizados en bicicleta por la ciudad, ilustrando con mapas de calor los carriles bici más empleados.

4. Metodología del Proyecto

4.1. Fuente Datos

Los datos de este proyecto se han extraído del servicio de [Datos Abiertos del Ayuntamiento de San Sebastián](#). En concreto, se ha obtenido la [red de carriles bici](#) (.kml), los [aparcamientos públicos de bicicletas](#) de la ciudad (kml), y la [disponibilidad de bicicletas](#) en tiempo real (.json)

unixtime	stations_json
1530356436	{{"numero_estacion": "1", "numero_bases": "18", "bases_libres": "6", "bases_enganchadas": "11", "activada": "1", "no_operativa": "0"},
1530356677	{{"numero_estacion": "1", "numero_bases": "18", "bases_libres": "4", "bases_enganchadas": "13", "activada": "1", "no_operativa": "0"},

Los usuarios de dBizi pagan en función del uso que hacen del servicio. Obviamente, el sistema informático de dBizi registra en algún punto los tiempos de salida y llegada de cada usuario (desconozco si las estaciones también se registran). Sin embargo, esta información personal de cada usuario es inaccesible para el público. Cabe destacar que en este proyecto se ha realizado una estimación de los viajes realizados empleando únicamente el número de bases disponibles en cada estación, y no datos personales de los usuarios.

4.2. Adquisición Datos

A pesar de contar con la disponibilidad de bicicletas en tiempo real, el servicio de datos abiertos no ofrece la posibilidad de acceder a datos históricos. Es por ello que, con el objetivo de estudiar la viabilidad de este proyecto, se ha desarrollado una simple plataforma de adquisición de datos en una [Raspberry Pi](#). Este plataforma comprueba, con una frecuencia minatural, si los datos a tiempo real han sufrido algún cambio, en cuyo caso, dicha información se almacena en una base de datos [PostgreSQL](#) con formato JSON. Con esta plataforma se han almacenado datos durante dos semanas de Junio de 2018.

4.3. Análisis Datos

Tras la obtención de datos, se ha realizado un análisis exploratorio y de limpieza en Python. Por un lado, se han extraído las variables más interesantes para este estudio: longitud y latitud de las estaciones, y disponibilidad de bases libres (u ocupación de bases enganchadas). En este sentido, se ha visto que la suma de bases libres y bases enganchadas no se mantiene constante al número de bases totales en cada estación. A pesar de que pueda haber un retraso en la actualización de los datos, no se han visto un patrón aclaratorio.

4.4. Algoritmo Estimación Rutas

Los datos de los que parte este algoritmo son: En un tiempo t , se ha registrado un cambio en el número de bases disponibles k de una estación s . Si dicho cambio es positivo, se considera que aumenta el número de bases libres.

Con dichos datos, y considerando que un usuario no volverá a la misma estación de la que partió, se construye el conjunto de todos los viajes posibles X :

$$x_i: s_A, t_U \rightarrow s_B, t_V$$

siendo s_A, s_B las estaciones origen y destino, y t_U, t_V los tiempos de salida y llegada. Con ello, considerando un tiempo acotado (servicio diario de dBizi de 06:30 a 23:00 horas), se ha resuelto el siguiente problema de optimización lineal entera por medio de [IBM ILOG CPLEX](#):

$$\min \sum_i c_i * x_i$$

A cada viaje x_i se le asigna un 1 si se realiza y un 0 en caso contrario. El coste de un viaje c_i se calcula como la diferencia entre la duración estimada de un viaje en bicicleta entre las estaciones s_A y s_B y el tiempo $t_U - t_V$. Esta duración se ha estimado con la API de [GraphHopper](#).

La minimización está sujeta a:

$$a) \sum_j x_j = k_{s,t} \forall s \in X$$

Siendo x_j cualquier viaje que comience en s , en un tiempo t , y finalice en cualquier otra estación que no sea s , en un tiempo posterior a t .

$$b) \sum_j x_j = -k_{s,t} \forall s \in X$$

Siendo x_j cualquier viaje que finalice en s , en un tiempo t , y haya comenzado en cualquier otra estación que no sea s , en un tiempo anterior a t .

4.5. Visualización

Para la visualización se ha empleado la librería [Leaflet.js](#), que mediante un módulo de visualización temporal ([TimeDimension](#)), ha permitido incluir los viajes de cada bicicleta sobre el mapa de San Sebastián. Por otro lado, las gráficas de frecuencias y estadísticas del análisis exploratorio se han obtenido por medio de [Matplotlib](#) de Python.

4.6. Aplicación Web

Por último, la [aplicación web](#) se ha desarrollado mediante el servicio [Firebase de Google](#). Actualmente la web solo ilustra el proyecto y el concepto de la aplicación.

