



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **APLICACIÓN WEB DE ESTUDIO DE MOVILIDAD EN BICICLETA PARA LA CIUDAD DE VALENCIA**

## **Trabajo Fin de Máster**

**Máster Universitario en Gestión de la Información**

**Autor:** Benjamin Arroquia Cuadros

**Tutor:** Nuria Lloret Romero

Curso 2016 - 2017



# Agradecimientos

A todos aquéllos que han aportado sus consejos tanto técnicos como personales.

A mis amigos en especial a Marta, Pablo y Álvaro por su ayuda.

Al equipo de ViceBici con el que logramos ganar el primer Datathon UPV.

A mi familia, en especial a mis padres y hermanas por dar siempre un sentido a las cosas.

Y a los profesores de la Universidad Politécnica de Valencia por todo lo que me han enseñado, en especial a Ángel y Núria por el compromiso y su vocación.

Gracias



# Resumen

La movilidad urbana es una de las cuestiones más recurrentes para los habitantes de una ciudad, ya que supone un impacto directo en la satisfacción de los ciudadanos con su entorno. El ecosistema existente sobre la movilidad urbana es muy complejo, ya que hay muchos factores a tener en cuenta y tienen que coexistir de manera coherente e integradora. En los últimos años, Valencia ha experimentado un auge en el uso de la bicicleta como una alternativa a los medios tradicionales de movilidad. Este hecho ha generado un impacto en la ciudad tanto en el plano político como económico, y que motiva a estudiar y entender este recurso en la ciudad de Valencia mediante el uso de técnicas propias de las TICs.

Los nuevos servicios web o repositorios de datos abiertos, creados por las instituciones públicas, permiten realizar actividades derivadas como el estudio de los fenómenos disponibles o para generar nuevos modelos de negocio basados en la información.

La aplicación web de estudio de movilidad en bicicleta realiza un tratamiento de datos mediante diferentes técnicas propias de la gestión de información, de este modo se consigue conocimiento derivado de la información y disponible para que el ciudadano obtenga conclusiones aplicando su propio análisis. La aplicación cuenta con diferentes funcionalidades para la visualización con el objetivo de hacer entender este fenómeno de una forma fácil y sencilla.

## Palabras clave

Visualización, Minería de datos, Análisis espacial, Movilidad urbana, Datos abiertos, API, Diseño web, Bicicleta.

# Summary

Urban mobility is one of the most recurrent issues for city inhabitants, since it has direct impact on the citizens' satisfaction with their environment. The existing ecosystem on urban mobility is very complex, as there are many factors to be taken into account and have to coexist in a coherent and inclusive way. In recent years, the city of Valencia has experienced a boom in bicycle's use as an alternative to traditional mobility means. This fact has generated an impact in the city both politically and economically which motivates to study and understand this resource in the city of Valencia by using ICTs techniques.

New web services or open data repositories, created by public institutions, allow making derivative activities such as the study of available phenomena or generating new business models based on information.

This web application for cycling mobility analysis performs data treatment by using different information management techniques, so information derivative knowledge is provided and available for citizens, who can get conclusions by applying their own analysis. This application has different visualization functionalities with the objective of making understand this phenomenon on an easy and simple way.

## Key words

Visualization, Data Mining, Spatial analyst, Urban mobility, Open data, API, Web design, Bicycle

# Resum

La mobilitat urbana és una de les qüestions més recurrents per als habitants d'una ciutat, ja que suposa un impacte directe en la satisfacció dels ciutadans amb el seu entorn. L'ecosistema existent sobre la mobilitat urbana és molt complex, ja que hi ha molts factors a tindre en compte i han de coexistir de manera coherent i integradora. En els últims anys, València ha experimentat un auge en l'ús de la bicicleta com una alternativa als mitjans tradicionals de mobilitat. Aquest fet ha generat un impacte en la ciutat tant en el pla polític com econòmic, i que motiva a estudiar i entendre aquest recurs en la ciutat de València per mitjà de l'ús de tècniques pròpies de les TICs.

Els nous servicis web o repositoris de dades obertes, creats per les institucions públiques, permeten realitzar activitats derivades com l'estudi dels fenòmens disponibles o per a generar nous models de negoci basats en la informació.

L'aplicació web d'estudi de mobilitat amb bicicleta realitza un tractament de dades per mitjà de diferents tècniques pròpies de la gestió d'informació, d'aquesta manera s'aconsegueix coneixement derivat de la informació i disponible per a que el ciutadà obtinga conclusions aplicant la seua pròpia anàlisi. L'aplicació compta amb diferents funcionalitats per a la visualització amb l'objectiu de fer entendre aquest fenomen d'una forma fàcil i senzilla.

## Paraules clau

Visualització, Minería de dades, Anàlisi espacial, Mobilitat urbana, Dades obertes, API, Disseny web, Bicicleta.



# Índice

<b>1</b>	<b>Introducción.....</b>	<b>1</b>
<b>2</b>	<b>Objetivos.....</b>	<b>2</b>
<b>3</b>	<b>Antecedentes.....</b>	<b>3</b>
3.1	Las ciudades como marco de estudio.....	3
3.2	Movilidad urbana en bicicleta.....	4
3.3	Smart City.....	7
3.4	Ciudad de Valencia.....	10
3.5	Estado del arte.....	13
3.5.1	<i>Fuentes de información.....</i>	<i>13</i>
3.5.2	<i>Aplicaciones con datos abiertos.....</i>	<i>14</i>
<b>4</b>	<b>Diseño del sistema y componentes.....</b>	<b>16</b>
4.1	Definición del sistema.....	16
4.2	Análisis funcional del sistema.....	17
4.2.1	<i>Requisitos no funcionales.....</i>	<i>18</i>
4.2.2	<i>Requisitos funcionales.....</i>	<i>18</i>
4.3	Estudio de alternativas.....	19
4.3.1	<i>Base de datos.....</i>	<i>19</i>
4.3.2	<i>Procesamiento de datos.....</i>	<i>20</i>
4.3.3	<i>Framework web.....</i>	<i>20</i>
4.3.4	<i>Web mapping.....</i>	<i>21</i>
4.3.5	<i>Gráficas.....</i>	<i>21</i>
4.4	Diseño de solución tecnológica.....	22
4.4.1	<i>Arquitectura del sistema.....</i>	<i>22</i>
4.4.2	<i>Capa datos.....</i>	<i>23</i>
4.4.3	<i>Capa de negocio.....</i>	<i>24</i>
4.4.4	<i>Capa de cliente.....</i>	<i>25</i>
<b>5</b>	<b>Índices de movilidad urbana para bicicletas.....</b>	<b>28</b>
5.1	Datos sobre movilidad urbana.....	28
5.2	Unidades de comparación.....	30
5.3	Índices de estudio.....	31
5.3.1	<i>Porcentaje de carril bici por vía urbana.....</i>	<i>31</i>
5.3.2	<i>Población de un barrio por borneta de estación de Valenbisi.....</i>	<i>32</i>
5.3.3	<i>Distancia media del carril bici a los portales.....</i>	<i>33</i>
5.3.4	<i>Máxima distancia en un barrio entre portales y carriles bici.....</i>	<i>34</i>
<b>6</b>	<b>Desarrollo de la aplicación web.....</b>	<b>35</b>
6.1	Entorno de desarrollo.....	35
6.1.1	<i>Sistema Gestor de Base de Datos.....</i>	<i>36</i>
6.1.2	<i>Framework web.....</i>	<i>37</i>

6.1.3 Repositorio Bitbucket.....	38
6.2 Gestión y tratamiento de la información.....	38
6.2.1 Procedimiento de obtención de conocimiento.....	38
6.2.2 Modelo de datos.....	39
6.2.3 Importación de datos.....	40
6.2.4 Mantenimiento y procesamiento de datos.....	41
6.2.5 Proceso de transformación de datos.....	42
6.2.6 Resultados del análisis.....	49
6.3 Desarrollo web.....	51
6.3.1 Diseño de aplicación web.....	51
6.3.2 Desarrollo servidor.....	51
6.3.3 Desarrollo cliente.....	57
6.4 Despliegue de la aplicación.....	64
<b>7 Conclusiones.....</b>	<b>67</b>
7.1 Líneas de mejora.....	67
<b>8 Bibliografía.....</b>	<b>69</b>
<b>9 Anexos.....</b>	<b>71</b>
9.1 Tabla de resultados.....	71

# Índice de figuras

Figura 1: División submunicipal de barrios y distritos en el término municipal de Valencia. Elaboración propia. Fuente [8].....	4
Figura 2. Gráfica de longitudes totales de carriles bici en Valencia, Sevilla, Barcelona y Madrid. Elaboración propia. Datos [8], [9], [10], [11].....	5
Figura 3. Diagrama de Smart City. Fuente [12].....	7
Figura 4: Fiware iniciativa europea para aplicaciones de Smart Cities. Fuente luissenlabs.com.....	8
Figura 5: Mapa de situación de la ciudad de Valencia. Elaboración propia. Datos arcgis.com e IGN.....	10
Figura 6: Mapa de coropletas de la población por distritos en la ciudad de Valencia. Elaboración propia. Datos [8].....	11
Figura 7: Gráfica sobre tipos de carril bici en Valencia con sus longitudes. Fuente [8].....	12
Figura 8: Dashboard del portal de datos abiertos de Valencia. Fuente [8].....	14
Figura 9: Índices de polución del proyecto airVLC. Fuente <a href="http://safe-tools.dsic.upv.es/airvlc/index.php">http://safe-tools.dsic.upv.es/airvlc/index.php</a> .....	15
Figura 10: Bicimetro, visor web para localización de rutas para bicicleta. Fuente <a href="http://bicimetro.plumtic.es/">http://bicimetro.plumtic.es/</a> . .....	15
Figura 11: Estructura de capas del sistema. Elaboración propia.....	16
Figura 12: Diagrama de capas del proyecto. Elaboración propia.....	17
Figura 13: Alternativas tecnológicas. Fuentes <a href="http://aws.com">aws.com</a> , <a href="http://mongodb.com">mongodb.com</a> , <a href="http://postgis.us">postgis.us</a> , <a href="http://python.org">python.org</a> , <a href="http://fullstackpython.com">fullstackpython.com</a> , <a href="http://leafletjs.com">leafletjs.com</a> , <a href="http://ubuntu.org">ubuntu.org</a> .....	19
Figura 14: Arquitectura del sistema para la aplicación web. Elaboración propia.....	22
Figura 15: Distribución de esquemas para los datos del proyecto. Elaboración propia.....	23
Figura 16: GeoJSON como formato para el servicio web de datos. Elaboración propia.....	25
Figura 17: Muestra de ajuste responsive de la aplicación web con tres tipos de anchos de pantalla (1890px, 930px, 630px). Elaboración propia.....	26
Figura 18: Mapa de coropletas del porcentaje de carril bici por barrio de Valencia. Elaboración propia.....	27
Figura 19: Estación de Valenbisi en avenida Tarongers compuesto por bornetas donde se estacionan las bicicletas. Fuente <a href="http://eldiario.es">eldiario.es</a> .....	29
Figura 20: Representación de las islas catastrales, secciones censales y barrios en Valencia. Elaboración propia.....	30
Figura 21: Intersección espacial de dos geometrías, la primera tipo polígono (dimensión 2) la segunda tipo línea (dimensión 1). Resultado de la intersección son líneas que intersectan con el polígono. Elaboración propia.....	31
Figura 22: Cálculo del porcentaje de carril bici por barrio. Se muestra una tabla con los datos de los barrios contiguos a Benimaclet. Elaboración propia.....	32
Figura 23: Índice de personas por borneta por barrio. Se muestra en el mapa el barrio de la petxina con los datos de manzanas y su población junto con las estaciones de Valenbisi. Elaboración propia.....	33

Figura 24: Representación de los portales con una rampa de color en relación a la distancia euclídea al carril bici más cercano. Elaboración propia.....	34
Figura 25: Cliente psql de PostgreSQL. Elaboración propia.....	36
Figura 26: Repositorio del proyecto en Bitbucket. Fuente bitbucket.org.....	38
Figura 27: Tabla de estadísticas de barrios. Elaboración propia.....	40
Figura 28: Migración de datos desde QGIS a PostgreSQL. Elaboración propia.....	41
Figura 29: Modelo de generación de datos en el esquema metrics a partir de datos obtenidos de las distintas ciudades. Elaboración propia.....	42
Figura 30: Aplicación de la función ST_Intersection a la capa de carril bici con barrios. A la izquierda se aprecia como las líneas son continuas sin separación entre barrios, a la derecha se puede ver según el color el carril bici en función del barrio en el que se ubica. Elaboración propia.....	43
Figura 31: Sentencias SQL para división por barrio de los carriles bici y la agrupación por barrio. Elaboración propia.....	43
Figura 32: Ejemplo de distribución de los portales y un carril bici. Elaboración propia.....	44
Figura 33: Multiplicación de tablas con Left Join en PostgreSQL, lo cual relaciona cada portal con todos los carriles bici. Elaboración propia.....	45
Figura 34: Obtención de la distancia mínima de un portal al carril bici. Elaboración propia.....	45
Figura 35: Mapa con los puntos representando los portales con una rampa de color en función de la distancia al carril bici más cercano. En el centro se observa una zona rodeada por carriles bici y como distancias desde el centro son mayores. Elaboración propia.....	45
Figura 36: Sentencias SQL para el cálculo de distancia máxima y media por barrio mediante el uso de funciones de PostGIS. Elaboración propia.....	46
Figura 37: Gráfica de la distribución de los valores (eje de ordenadas) que tiene cada barrio (eje abscisa) por los cuatro índices calculados. Elaboración propia.....	47
Figura 38: Tabla de estadísticas de los índices calculados. Elaboración propia.....	48
Figura 39: Histograma con valores acumulados en 10 conjuntos (bins). Elaboración propia.....	48
Figura 40: Histograma con valores acumulados en 10 conjuntos (bins) corrigiendo los outliers. Elaboración propia.....	49
Figura 41: Fórmula de normalización máximo mínimo.....	49
Figura 42: Mapa de coropletas de los barrios de Valencia con las calificaciones del análisis de movilidad urbana. Elaboración propia.....	50
Figura 43: Estructura de ficheros de una aplicación Django. Elaboración propia.....	53
Figura 44: Diseño de mockups de las páginas con su correspondiente dirección URL y lo que se va a mostrar. Elaboración propia.....	54
Figura 45: Página principal del proyecto. Se aprecia el encabezado las secciones en las que se compone la página inicial. Elaboración propia.....	55
Figura 46: Diseño dashboard. Elaboración propia.....	55

Figura 47: Dashboard.html con la definición de bloques “container” y “jss”. El resto de código HTML y de etiquetas script se heredan en los ficheros que hagan uso de la función extend. Elaboración propia.....	56
Figura 48: Composición de las páginas referentes al dashboard. Elaboración propia.....	58
Figura 49: Página KPIs información sobre índices utilizados para el análisis de movilidad. Elaboración propia.....	58
Figura 50: Mapa de selección de barrios y cuadro de datos del barrio seleccionado. Elaboración propia. ....	59
Figura 51: Clase HighCharts de control de los elementos de que están seleccionando en el mapa y gráficas. Elaboración propia.....	60
Figura 52: Búsqueda de un barrio (parte superior) selección del barrio Ruzafa (parte inferior). Elaboración propia.....	60
Figura 53: Índices con máximos y mínimos. Elaboración propia.....	61
Figura 54: Mapa con los barrios seleccionados en azul que tienen los datos en las gráficas. Elaboración propia.....	61
Figura 55: Gráfica de barras (izquierda) compara las distancias máximas y mínimas del carril bici más cercano en cada barrio. Gráfica polar (derecha) se representa la cantidad de personas por borneta de carril bici. Elaboración propia.....	62
Figura 56: Gráfica de tarta comparando la longitud del carril bici. Muestra de desactivar elementos de una gráfica (izquierda). Elaboración propia.....	62
Figura 57: Tablas con el ranking del porcentaje de carril bici. Elaboración propia.....	63
Figura 58: Mapa de coropletas del porcentaje de carril bici en Valencia. Elaboración propia.....	64
Figura 59: Mapa de coropletas del índice de población por bornetas de Valenbisi. Elaboración propia.....	64
Figura 60: Acceso a la base de datos del servidor en remoto desde el cliente pgAdmin usando el puerto 3333. Previamente se ha debido generar el túnel de conexión. Elaboración propia.....	65
Figura 61: Diagrama de flujo de la configuración del servidor y las conexiones de los distintos módulos. Se ha utilizado un terminal para realizar las conexiones al servidor y crear un túnel ssh. Elaboración propia. ....	66

# 1 Introducción

La movilidad urbana es una de las cuestiones más recurrentes para los habitantes de una ciudad, ya que supone un impacto directo en la satisfacción de los ciudadanos con su entorno. El ecosistema existente en este ámbito es muy complejo, ya que hay muchos factores a tener en cuenta y tienen que coexistir de manera coherente e integradora.

En los últimos años, las ciudades españolas y de manera más significativa Valencia, han experimentado un auge en el uso de la bicicleta como una alternativa a los medios tradicionales de movilidad urbana. Este hecho ha generado un impacto tanto en el plano político como económico de la ciudad, que motiva a estudiar y entender este recurso mediante el uso de técnicas propias de las TICs.

Este fenómeno se alinea temporalmente con otro de carácter tecnológico denominado “Smart City”. En él, se generan unas directivas gubernamentales enfocadas a la adaptación de los modelos tradicionales de gestión urbana hacia el uso de las TIC para la mejora de la eficiencia en materia de gestión urbana; de este modo aumentan las capacidades para crear conexiones entre los distintos agentes que componen el entorno urbano y también, para ofrecer soluciones adaptadas a las diferentes casuísticas que se puedan dar.

En el marco de las ciudades inteligentes, los datos y la gestión de la información son un factor clave. Los repositorios de datos abiertos, creados por las instituciones públicas, permiten realizar actividades derivadas como el estudio de los fenómenos y recursos disponibles o generar nuevos modelos de negocio basados en la información.

Este conjunto de datos, llamado “Open Data”, es publicado y puede consultarse libremente, haciendo posible el análisis de fenómenos que tienen que ver con espacios relacionados con la actividad de los ciudadanos. Es por eso que estos datos generan un gran interés, pero en cierta medida no llega al ciudadano ya sea por motivos de inexperiencia en el tratamiento de datos o por el coste temporal que supondría el análisis de los mismos.

La aplicación web de estudio de movilidad en bicicleta Vicebici, realiza un tratamiento de datos mediante diferentes técnicas propias de la gestión de información, de este modo se consigue conocimiento derivado de la información y disponible para que el ciudadano obtenga conclusiones aplicando su propio análisis. La aplicación cuenta con diferentes funcionalidades para la visualización con el objetivo de hacer entender este fenómeno de una forma fácil y sencilla.

Vicebici pretende acercar, tanto a los ciudadanos como técnicos, datos referentes a la movilidad urbana en bicicleta mediante el uso de visualizaciones interactivas.

## **2 Objetivos**

El objetivo del presente proyecto es realizar un análisis sobre la movilidad urbana en bicicleta en la ciudad de Valencia. Este estudio se basa en dar respuesta a las incógnitas que surgen tras el auge actual de la movilidad en bicicleta y su tendencia en el uso por parte de los ciudadanos y turistas en Valencia.

Para realizar este estudio es necesario analizar los diferentes agentes que intervienen en la movilidad urbana, de forma que las conclusiones que se generen tras el estudio esté adaptado a la realidad actual. Por tanto, se establece una metodología de análisis basada en datos que puedan combinar tanto la componente espacial como las características propias de los elementos a analizar.

Se establece como fuente de datos aquéllas que tengan que ver con repositorios abiertos y/o oficiales, conservando el carácter libre y colaborativo del estudio, y fundamentándolo en la interoperabilidad y seguridad de las fuentes oficiales.

El análisis de movilidad tiene como reto que cualquier ciudadano pueda conocer con detalle y rigor el estado de la movilidad urbana en bicicleta y obtener sus propias conclusiones mediante datos objetivos. En esta línea se opta como canal de comunicación las tecnologías web por su gran accesibilidad y las posibilidades que ofrecen como interfaz de visualización de datos.

El estudio de movilidad urbana en la ciudad de Valencia genera unos datos, que previamente han sido procesados y dar un valor añadido a los datos dotándolos de conocimiento, para obtener unos resultados que permita identificar las zonas con mejores dotaciones relacionadas a la movilidad en bicicleta.

Para la realización de este proyecto se plantean unos objetivos específicos que concretan el objetivo general en:

- Generar una metodología de análisis basada en datos espaciales.
- Reutilización de datos abiertos y oficiales.
- Realizar análisis sobre movilidad urbana en bicicleta de Valencia.
- Obtención de resultados sobre el análisis movilidad urbana en bicicleta.
- Publicación datos mediante una aplicación web.

## 3 Antecedentes

### 3.1 Las ciudades como marco de estudio

En el presente proyecto, cuyo tema principal es la movilidad en bicicleta, se define como marco de estudio las ciudades, puesto que la movilidad en bicicleta supone actualmente un medio habitual de transporte para muchas personas.

Se escoge el contexto urbano debido al aumento del uso de este medio como alternativa a otros. Es en este contexto donde actualmente está habiendo políticas de fomento del uso de este medio, debido a muchos factores que provocan descontento en la actividad urbana por parte de los ciudadanos.

Las ciudades son un entorno en el cual no sólo hay grandes densidades de población, sino que existe una actividad económica que supone una recepción mucho mayor de personas por movilidad residencial como uno de los procesos urbanos, provocando ciertas problemáticas como congestión de circulación, masificación o aumento de contaminación atmosférica y acústica, entre otras. Estos núcleos de actividad socio-económico donde se encuentran sistemas complejos que inevitablemente interaccionan y pueden condicionar la vida de los ciudadanos en tanto a sus necesidades como su potencial económico, es por eso que se necesita adaptar las dotaciones e infraestructura de forma adecuada.

La definición de ciudad viene dada en función de la población de un asentamiento y la densidad que representa su distribución en el centro urbano. De esta forma se definen varias casuísticas debido a las dimensiones que se pueden llegar a dar (en concreto la tipología deriva de la comisión Europea y a su vez de la OCDE) definiéndose ciudades y conurbaciones. Es en este sentido donde se definen dos niveles que forman parte de las ciudades: nivel supramunicipal y submunicipal [2].

- Supramunicipal, también llamados Área Urbana Funcional engloba todas las zonas de un municipio en el cual un porcentaje de sus habitantes se desplaza al núcleo central por motivos de trabajo.
- Submunicipal también llamados distritos, barrios o de forma técnica SCD (Sub-City Districts). En el cual se engloban poblaciones entre 2.000 y 40.000 habitantes. El INE junto con los ayuntamientos definen estas subdivisiones para municipios de 250.000 habitantes.

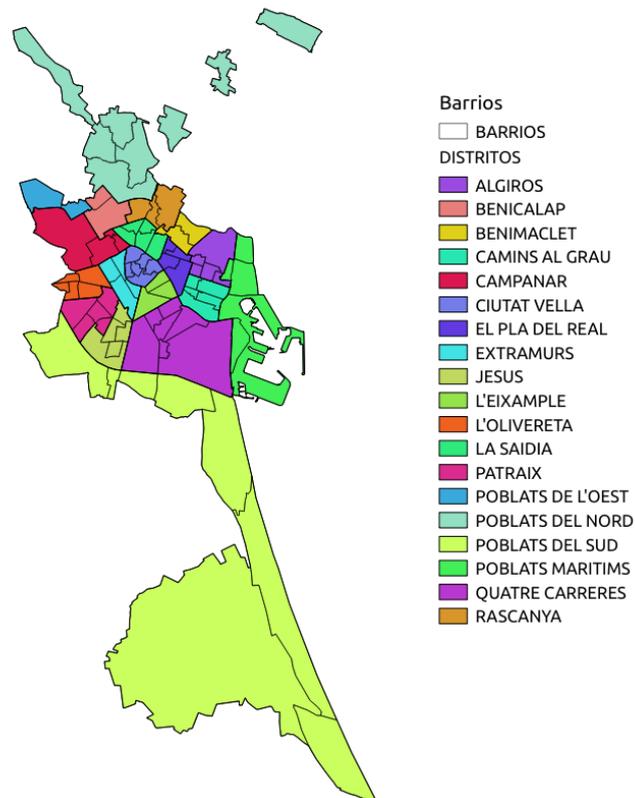


Figura 1: División submunicipal de barrios y distritos en el término municipal de Valencia. Elaboración propia. Fuente [8].

Esta división submunicipal (Figura 1) se establece como referencia para la creación de índices de movilidad del presente proyecto.

## 3.2 Movilidad urbana en bicicleta

La movilidad en las ciudades es un tema que preocupa mucho a los ciudadanos, especialmente los que residen en grandes ciudades como Madrid, Barcelona o Valencia (entre otros). Es en estas ciudades donde se concentra mayor tráfico de automóviles (siendo éstos el medio de transporte mayoritario) provocando un malestar en la ciudadanía, debido entre otras causas a:

- Tráfico y congestión en la circulación
- Contaminación acústica
- Contaminación atmosférica
- Limitación de aparcamiento

Dado la tendencia de incremento de la población en las grandes ciudades, esto conlleva a una mayor necesidad de circulación por parte de los ciudadanos. Esta tendencia tiene que estar ligada al uso de transportes alternativos que permitan una evolución sostenible de las ciudades como son los medios de transporte público, vehículos eléctricos o, como es el caso de estudio, la movilidad en bicicleta. En España

ha habido un incremento de la venta de bicicletas y, en términos globales, las ventas superaron a las de automóviles en 2015 [3] y sobre todo bicicletas de ciudad, y es debido al aumento de la movilidad.

La movilidad supone el desplazamiento de las personas sin importar el medio que utilicen para dicho fin [4], es importante no confundir la movilidad con el transporte, así como de la circulación. Y es en las ciudades donde se genera el cambio de modelo tradicional de movilidad hacia uno económico, ecológico, saludable y sostenible como es el de la bicicleta.

Para que exista un cambio en el modelo de movilidad, es necesario que la sociedad esté concienciada en esta problemática y que haya una evidencia de que exista la necesidad de los ciudadanos de dotaciones e infraestructuras que permitan el uso de la bicicleta. La circulación por una ciudad con este medio de transporte, en muchas ocasiones supone un riesgo ya que se deben compartir las vías con otros vehículos más pesados. Este hecho genera una necesidad de crear carriles o vías destinadas a la circulación de bicicletas, de esta forma el usuario gana más confianza y conlleva a un aumento del uso de este medio.

En muchas ciudades, la bicicleta supone un medio habitual de transporte, el claro exponente Europeo es Amsterdam con un uso del 60% de la población [5]. En España, las ciudades de Sevilla y Valencia se sitúan como las mejores valoradas para el uso de bicicleta como medio de transporte [6]. Esto ha supuesto un interés y por otro lado un pretexto para la mejora de los servicios e infraestructuras sobre circulación en bicicleta.

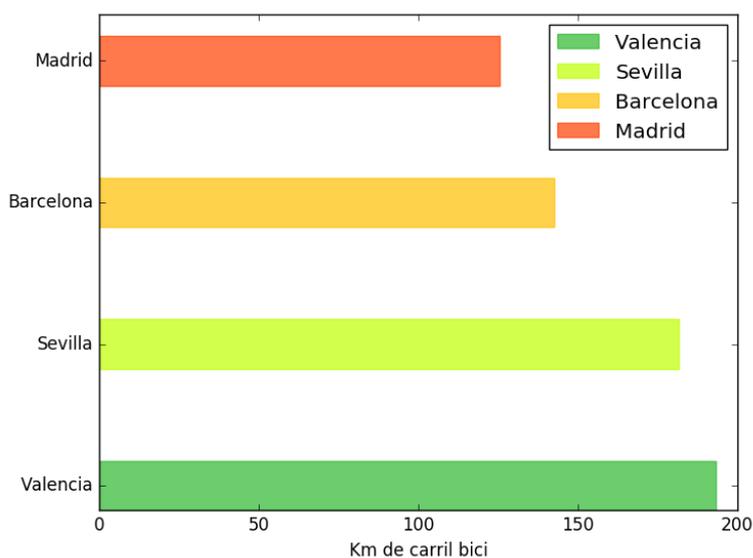


Figura 2. Gráfica de longitudes totales de carriles bici en Valencia, Sevilla, Barcelona y Madrid. Elaboración propia. Datos [8], [9], [10], [11].

Es importante tener en cuenta que el uso que se hace de la bicicleta en una ciudad puede ser por ocio o como medio de transporte. Esta distinción puede tener distintas lecturas ya que el uso de la bicicleta por ocio es muy probable que se concentre en días no laborables o por las tardes en un día normal, o también por el sector turístico. Sin embargo en las horas de máxima afluencia de tráfico es dónde probablemente el uso de

la bicicleta esté ligado al usuario que la utilice como medio de transporte. Aunque este análisis no es el propósito del presente proyecto, si lo es el estudio de cómo la ciudad está preparada para la movilidad en bicicleta por cualquier usuario.

Según el informe de la OCU, la valoración de una ciudad para los ciclistas se ha basado en unas encuestas sobre la situación geográfica, servicios de alquiler, aparcabicis, facilidad para acceder al transporte público, densidad y velocidad del tráfico y condiciones de las vías ciclistas. Pero los principales inconvenientes para el uso de bicicleta es tanto el excesivo tráfico como una red de carril bici escasa y mal conectada. No es casualidad la relación directa de los kilómetros de vías con el uso de la bicicleta en la ciudad como se observa en la Figura 2. Valencia y Sevilla se convierten en materia de carril bici en los referentes nacionales, más aún teniendo en cuenta que tanto Madrid (3.1 millones de hab.) y Barcelona (1.6 millones) disponen de una extensión mucho mayor que Valencia (0.78 millones hab.) y Sevilla (0.69 millones de hab.) [2].

### 3.3 Smart City

Una Smart City es un concepto que se basa en la conexión de los diferentes agentes que componen el ecosistema urbano para una gestión eficiente y sostenible de los recursos, y dotar de mayor participación a los ciudadanos. Los medios por lo que se convierte una ciudad en “Smart” pasan por el uso de nuevas tecnologías basadas en las TICs, ya sea por la organización de la información, la comunicación entre sensores o receptores de información, que mejoren la eficiencia de los servicios y las aplicaciones que dotan de participación a los ciudadanos.

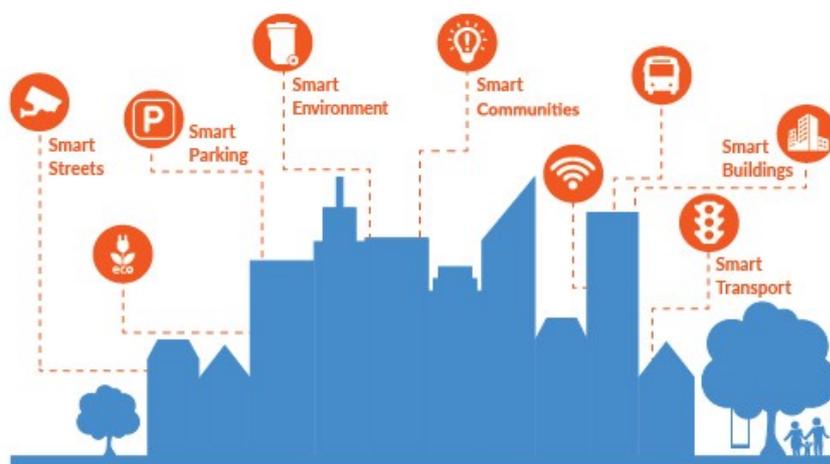


Figura 3. Diagrama de Smart City. Fuente [12].

La analogía más representativa de este concepto es un coche: el motor, los componentes que gestionan la coordinación de sus funciones y, por supuesto el combustible para su funcionamiento. Es este último el que representa a los datos, una ciudad inteligente sin datos de su propia actividad está ciega. Hasta hace unos años, las ciudades no disponían de las capacidades de la gestión de estos volúmenes (derivados a administraciones supramunicipales con más recursos). Pero en la nueva era de la información tanto el IoT como Big Data y una nueva generación de técnicos formados en la gestión de la información son determinantes en el sistema que define una Smart City.

La tendencia actual en esta materia pasa por la generación de un sistema de información capaz de generar servicios, haciendo posible que otros sistemas se puedan conectar a las fuentes de información o portales de datos. Es el caso de las aplicaciones móviles o web ya sean para uso de gestión municipal o para uso de los ciudadanos. En este sentido, también se crean iniciativas enfocadas a la transparencia de la gestión municipal como es el Open Data [13]:

- La Red Española de Ciudades Inteligentes (RECI) es una asociación de municipios con el objetivo de crear un modelo de eficiencia en la gestión de manera automatizada con el fin de reducir costes en la administración. Los principales colaboradores son la Fundación Española de Municipios y Provincias y Telefónica.

- datos.gob.es. Es un portal que sirve de repositorio que recoge todos los municipios con un servicio de datos abiertos.
- FIWARE. Iniciativa Europea de código abierto para la generación de estándares basados en tecnologías relacionadas con Smart Cities. Con esta iniciativa se pretende generar sistemas que recolectan, gestionen y den visibilidad a datos relacionados con la ciudad.



*Figura 4: Fiware iniciativa europea para aplicaciones de Smart Cities. Fuente luissenlabs.com.*

También se han realizado diferentes estudios y se han generado normas que pretenden homogeneizar criterios y establecer las bases para instaurar la base de una Smart City dentro del Plan Nacional de Ciudades Inteligentes. En este sentido España ha promovido la realización de unas normas para el desarrollo de ciudades inteligentes (15 normas UNE de AECNOR 2015) como es la de “Accesibilidad universal en las Ciudades Inteligentes” [20], a través del comité CTN 178 Ciudades Inteligentes el cual cuenta con cinco subcomités:

- Infraestructuras
- Indicadores y semántica
- Gobierno y movilidad
- Energía y medio ambiente
- Destinos turísticos inteligentes

Dentro del concepto de Smart City se sitúa el Open Data, que tiene como objetivo la reutilización de los datos que produce una ciudad mediante la publicación de los mismos con unos estándares de interoperabilidad. Para la obtención de conocimiento a partir de estos datos es necesario realizar un tratamiento apropiado. La minería de datos es un campo de la ciencia de datos que permite extraer un conocimiento que no se podrían obtener de manera directa. El análisis espacial supone otra técnica de tratamiento de datos para la obtención de conocimiento ya que gran parte de los datos de una ciudad pueden ser georreferenciados y la visualización en el espacio ayuda a identificar aspectos clave en la movilidad urbana.

En el presente proyecto se realizan distintas técnicas tanto de extracción de datos, tratamiento, explotación y análisis, junto con la aplicación web que dará soporte a la visualización de estos procesos y permita al usuario analizar la información.

Los datos susceptibles de análisis en este proyecto están relacionados con la movilidad en bicicleta. También se ha hecho un estudio de los datos disponibles para añadir al análisis aquellos que puedan estar relacionados con con la temática del proyecto y conocer el impacto que pueden tener en la movilidad urbana. Entre ellos está la distribución de la población dado que es un factor importante para conocer el equilibrio de la distribución de las dotaciones en el territorio. También otros actores que forman parte de la circulación junto con la bicicleta, como las vías para otros tipos de vehículos.

### 3.4 Ciudad de Valencia

La ciudad de Valencia se sitúa en el Este de España y es la capital de la Comunidad Valenciana y de la provincia de Valencia. La principal actividad económica es el sector de los servicios (84%), siendo el comercio minorista y mayorista el predominante [14]. En el sector de la industria (14%) destacan las actividades relacionadas con la fabricación de papel y las artes gráficas, madera y mueble, productos metálicos, calzado y confección. El clima de la ciudad es el mediterráneo con precipitaciones generalmente estacionales en Otoño y finales de Invierno [14].

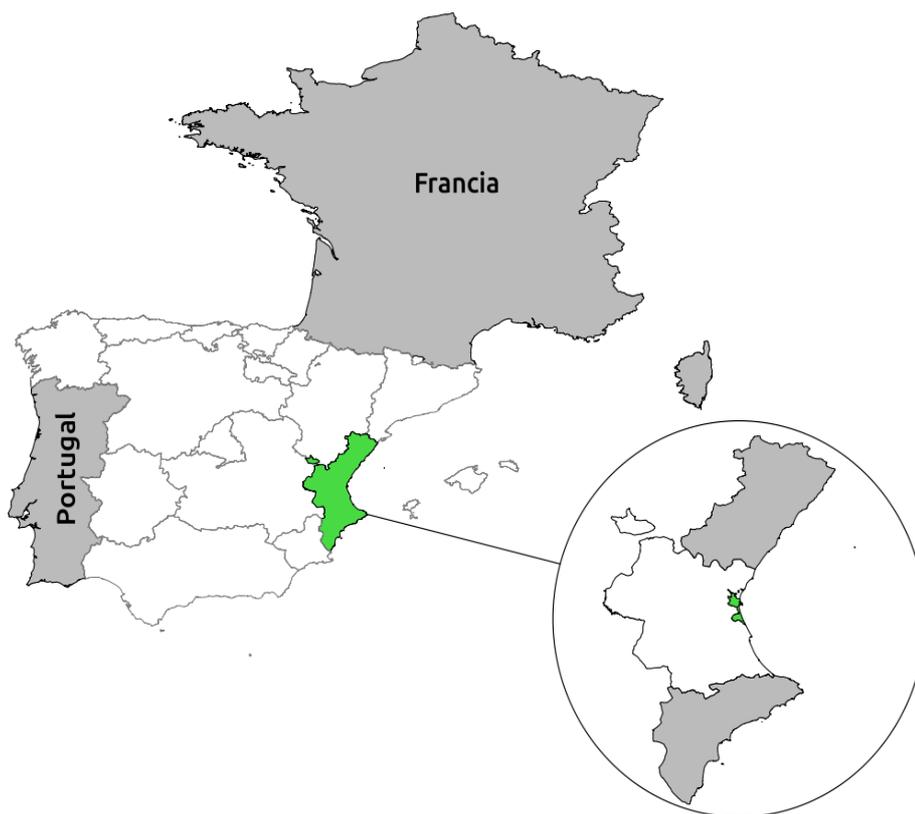


Figura 5: Mapa de situación de la ciudad de Valencia. Elaboración propia. Datos arcgis.com e IGN.

La población de Valencia es 796.779 habitantes [8] siendo el tercer municipio Español más poblado. Su distribución (Figura 5) es homogénea exceptuando los distritos de Poblats del Sur y Poblats del Nord que se sitúan en la periferia y en algunos casos separados notablemente del municipio.

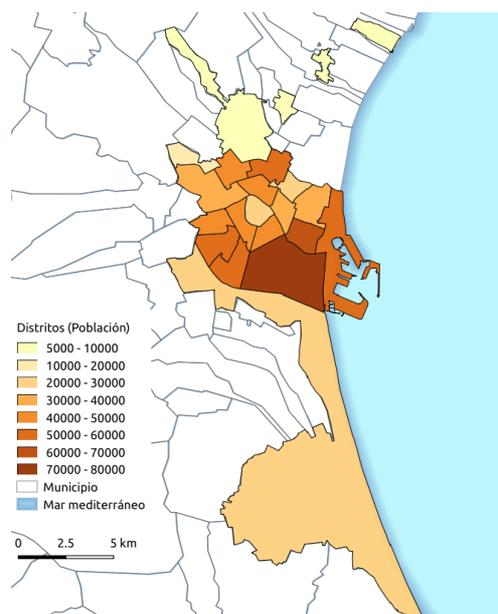


Figura 6: Mapa de coropletas de la población por distritos en la ciudad de Valencia. Elaboración propia. Datos [8].

La ciudad de Valencia en cuanto a movilidad urbana en bicicleta [7] tiene buena valoración sobre el conjunto nacional debido a diversos motivos:

- **Clima y orografía.** Clima templado con pocas precipitaciones y sus calles no presentan pendientes acentuadas que minimiza el esfuerzo de los usuarios en bicicleta.
- **Infraestructuras.** Valencia es la ciudad de España con mayor número de metros de carril bici y tiene un gran número de aparcamientos públicos para bicicletas. También dispone de una buena conexión en su red haciendo posible una buena circulación.
- **Transporte público.** Valencia cuenta con un sistema de transporte público basado en el autobús y metro. La Empresa Municipal de Transportes (EMT) cuenta con 60 líneas de autobús y 9 líneas de metro que también da servicio al área metropolitana. Esta última permite el transporte de bicicleta con algunas condiciones.
- **Servicio de alquiler público de bicicletas.** Valenbisi es el sistema de alquiler de bicicletas municipal e implementado y gestionado por JCDECaux y cuenta con 275 estaciones y 2750 bicicletas disponibles en las propias estaciones. Este sistema tiene una gran popularidad en la ciudad ya que se ha llegado a tener más de 90.000 usuarios anuales con picos de más de 600.000 viajes. Los usuarios de Valenbisi en 2016 fueron 45.026 y los viajes 479.000.
- **Medidas políticas.** En los últimos años los gobiernos están tomando medidas para favorecer el uso de la bicicleta que supone una ventaja para promover nuevas medidas en este ámbito.

El estudio de la OCU señala que “la ciudad del Turia es donde más ciclistas salen a la calle, ya sea ocasionalmente o de manera habitual. La bici constituye el principal medio de transporte para un 12% de los encuestados”.

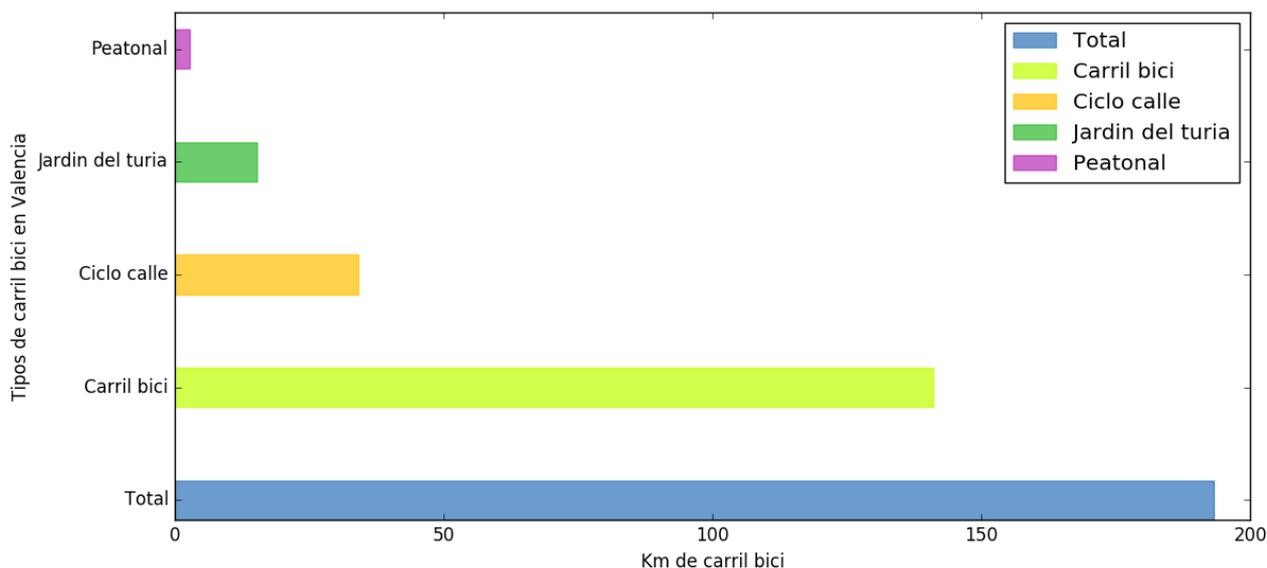


Figura 7: Gráfica sobre tipos de carril bici en Valencia con sus longitudes. Fuente [8].

Según lo comentado anteriormente, Valencia es la ciudad con más kilómetros de vías destinadas al carril bici (Figura 2), analizando los datos que proporciona el Portal de datos abiertos de Valencia se distinguen varias tipologías de carriles (Figura 7). El predominante es la tipología carril bici, vía destinada al tránsito exclusivo de bicicletas. En el resto de tipos, su beneficio podría resultar polémico para los usuarios de este transporte, ya que las ciclo calles comparten tránsito con vehículos motorizados, suponiendo un riesgo de accidente y la contaminación de los mismos es contraproducente a los beneficios de esta práctica. El Jardín del Turia, es un espacio ajardinado, pero en algunas situaciones incómodo para la circulación, ya que los accesos son limitados y puede penalizar en tiempo de recorrido, sumado al tránsito de peatones. Las zonas peatonales son espacios exclusivos para peatones, si se mezcla la circulación de bicicletas, puede provocar situaciones de riesgo.

Aun siendo un exponente en circulación de bicicletas, Valencia tiene muchos aspectos a mejorar en cuestión de calidad e implementación de vías ciclistas, así como la interconectividad con municipios contiguos según apuntan diversos estudios [6, 7].

## 3.5 Estado del arte

Actualmente los datos, y en especial los referentes a las ciudades, están cobrando un protagonismo nunca visto. Las posibilidades que tiene el tratamiento de datos gracias a diferentes técnicas como el análisis espacial, la minería de datos o el machine learning, permiten la obtención de análisis expertos que ayudan a comprender mejor los datos. En esta línea se han abierto paso diferentes campos de estudio que usan estas técnicas como: Business analytics, big data o geomarketing, entre otros.

Este proyecto se centra en las capacidades que tienen los datos referentes a las ciudades para analizar el estado de los agentes que intervienen en las ciudades y entender mejor sus relaciones. El objeto de análisis es la movilidad urbana y para ello es necesario conocer el estado actual en el campo de los datos abiertos y las diferentes actuaciones que se hayan podido llevar a cabo.

### 3.5.1 Fuentes de información

Uno de los propósitos del proyecto es la reutilización de datos y forman la base del proyecto para la llevar a cabo un análisis mediante diferentes técnicas. El estudio de las fuentes de información es el punto de partida del proyecto, ya que condiciona la extracción de los datos, su transformación y el alcance que pueda tener el análisis. Por este motivo es importante conocer las diferentes características de los conjuntos de datos para determinar la tecnología y metodologías que se deben aplicar para la obtención de información.

Las principales fuentes de información susceptibles a este proyecto son:

- Centro Nacional de Información Geográfica (CNIG) [16]. Es un organismo autónomo adscrito al Ministerio de Fomento a través de la Dirección General del Instituto Geográfico Nacional. En él se puede obtener la cartografía oficial del territorio nacional y diferentes productos.
- Instituto Nacional de Estadística (INE). El Instituto Nacional de Estadística es un organismo autónomo de carácter administrativo, con personalidad jurídica y patrimonio propio, adscrito al Ministerio de Economía, Industria y Competitividad a través de la Secretaría de Estado de Economía y Apoyo a la Empresa [18]. En este portal se pueden obtener diferentes conjuntos de datos, entre ellos datos de padrón municipal.
- Instituto Cartográfico Valenciano (ICV). Es un organismo de Conselleria d'Habitatge, Obres Públiques i Vertebració del Territori, y tiene como labor la generación de cartografía base y está suscrito como nodo de la red de Infraestructura de Datos Espaciales de España.
- Portal de datos abiertos del Ayuntamiento de Valencia [8]. En este portal se clasifican por temáticas diferentes conjuntos de datos como son las calles, carriles bici, distribución de la población, barrios, etc. Esta fuente ha sido la usada para el proyecto por su amplio catálogo y se ajusta a los parámetros del proyecto.

- datos.gob.es [13]. Es un portal de red.es perteneciente al ministerio de Hacienda y Función Pública, que reúne un catálogo de todas las instituciones públicas (municipios) con un repositorio de datos abiertos.

### 3.5.2 Aplicaciones con datos abiertos

Dentro de una ciudad inteligente, la apertura de datos es uno de los componentes que definen la conexión de la infraestructura y procesos internos con el entorno urbano y en definitiva con los ciudadanos. Este hecho se puede derivar diferentes medidas que se sustentan en el modelo de ciudades inteligentes y que tienen como objeto generar mayor actividad económica o desarrollar nuevas funcionalidades.

Un ejemplo son las APIs, nodos de conexión a aplicaciones que pueden integrarse a los datos que genera una ciudad. A día de hoy están surgiendo gran cantidad de proyectos que pretenden dar una respuesta a algunas preocupaciones de la ciudadanía mediante un enfoque denominado RISP (Reutilización de Información del Sector Público):

- Portal de datos abiertos de Valencia. Este portal cuenta con un gran catálogo de datos sobre distintas temáticas del municipio. Una de sus opciones también es un *dashboard* de algunos de los datos disponibles en el portal, dispone de gráficas (tarta, línea) o mapas con datos georreferenciables. Aunque hay muchas opciones resultan poco orientadas al usuario por ser poco claras y de difícil visualización (Figura 8).

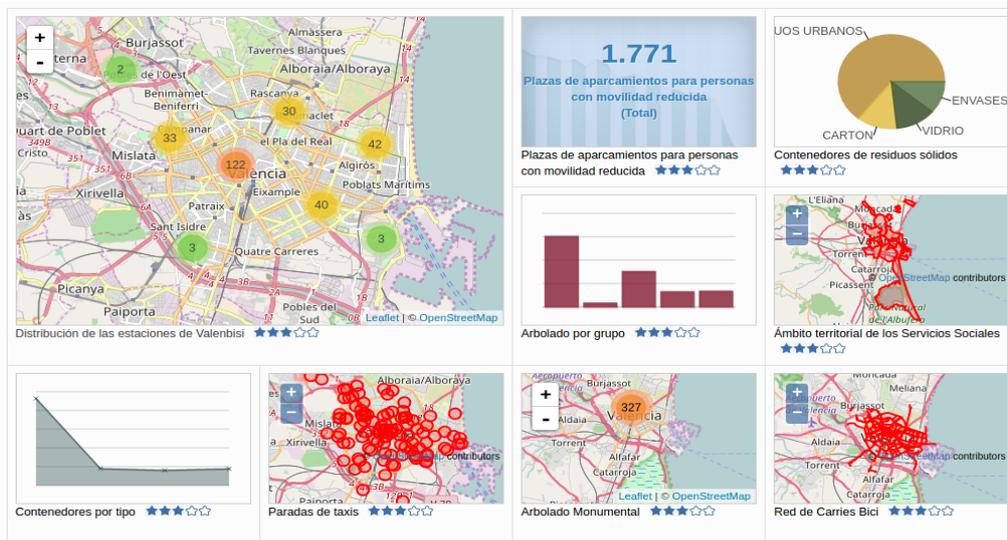


Figura 8: Dashboard del portal de datos abiertos de Valencia. Fuente [8].

- airVLC (Lidia Contreras, UPV). Estudio de predicción de la contaminación del aire en tiempo real. En el caso de este proyecto muestra una visualización compuesta por unos índices marcando unos límites a modo de KPIs y uso de mapas con Leaflet para mostrar resultados.

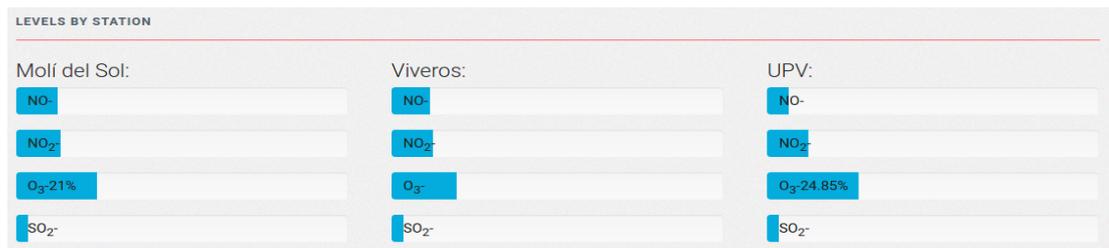


Figura 9: Índices de polución del proyecto airVLC. Fuente <http://safe-tools.dsic.upv.es/airvlc/index.php>.

- Bicimetro. Aplicación web y móvil para el cálculo de rutas de bici en la ciudad de Valladolid (Figura 10). La aplicación cuenta con un visor cartográfico hecho con OpenLayers 3 [19] y cuenta con algunas funcionalidades SIG como medición de áreas y líneas. Para el cálculo de rutas se selecciona en el mapa los puntos inicio y fin y se dibuja en el canvas una línea marcado la ruta propuesta.

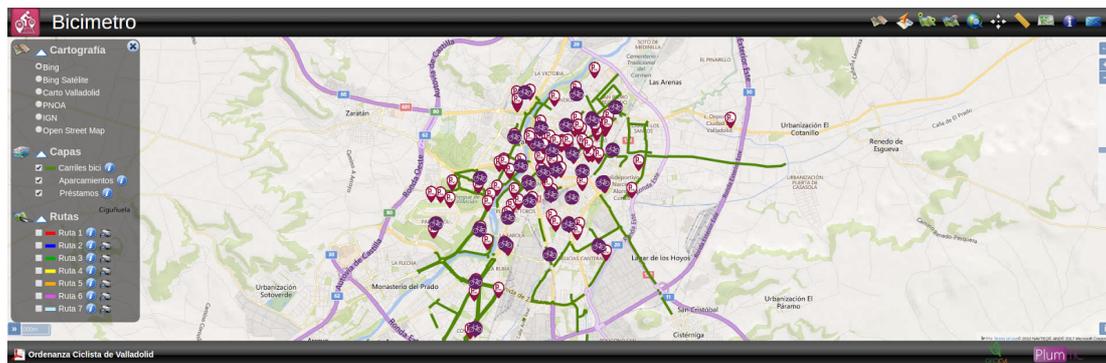


Figura 10: Bicimetro, visor web para localización de rutas para bicicleta. Fuente <http://bicimetro.plumtic.es/>.

## 4 Diseño del sistema y componentes

### 4.1 Definición del sistema

Para llevar a cabo los objetivos del proyecto, se define la estructura del sistema que sirve como marco de los componentes a desarrollar. El sistema tiene que adaptar los objetivos de publicación del análisis mediante una aplicación web. También el sistema se diseña para una gestión eficiente de la información y con capacidades adaptativas a otras aplicaciones.

El sistema está diseñado de forma modular basándose en una arquitectura por capas. Este diseño tiene como principal característica la encapsulación de cada una de las funciones de los módulos que componen el sistema de forma que estén aislados en cuanto al desarrollo. El aislamiento de los módulos tiene como principal ventaja el desarrollo sostenible de forma que las modificaciones no condicionen ninguno del resto de componentes del sistema.

La capas del sistema se definen como sigue (Figura 11):

- Capa cliente. Esta capa engloba todo desarrollo enfocado a la presentación de información o funcionalidades de interacción con el usuario. Las interfaces gráficas son las más comunes tanto para escritorio como para navegadores web.
- Capa negocio. Esta capa gestiona todos los procesos de transformación y generación de producto como funciones principales y se desarrollan los algoritmos principales del sistema.
- Capa datos. Los datos se encuentran en una capa en la que mantiene su persistencia e integridad.

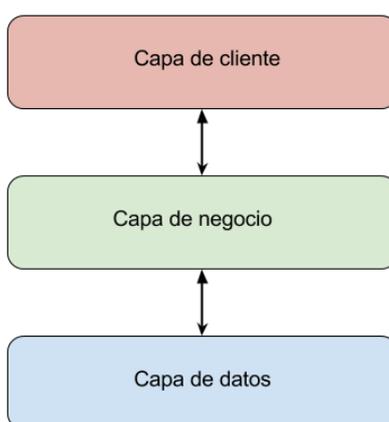


Figura 11: Estructura de capas del sistema. Elaboración propia.

Los módulos del sistema se definen como (Figura 12):

- Visualizador de mapas y dashboard. Este módulo tiene como objetivo mostrar datos del análisis al usuario en forma de mapas o dashboard.

- Framework web. La comunicación entre los datos y la capa de cliente se realiza mediante el módulo framework web. Éste establece un canal y los estándares de comunicación entre los diferentes módulos.
- Procesamiento de datos. Los datos que proceden de distintas fuentes se procesan de forma que habrá una transformación necesaria para generar analíticas.
- Base de datos geográfica. Los datos procedentes de las fuentes se almacenan en sistemas robustos y de forma que permita la indexación geográfica. Estos almacenarán los resultados de los procesos de la capa de negocio.

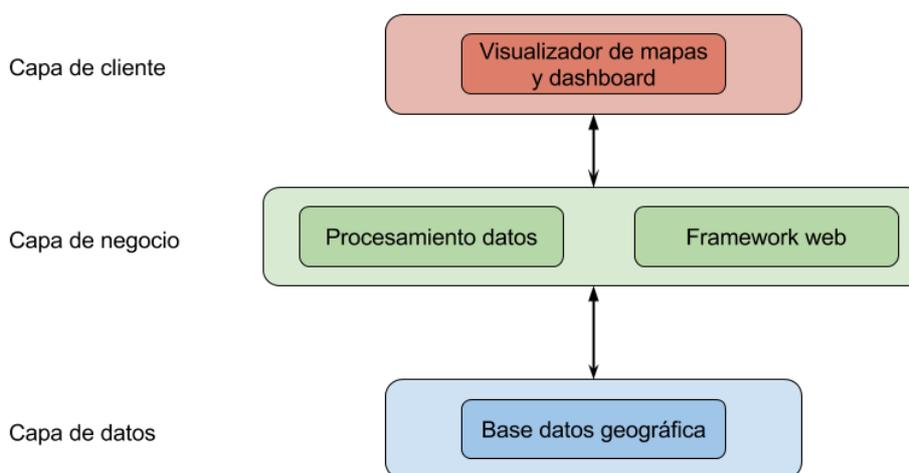


Figura 12: Diagrama de capas del proyecto. Elaboración propia.

## 4.2 Análisis funcional del sistema

El objetivo de este apartado es la definición de la arquitectura funcional del sistema, y así determinar el alcance de forma que se establece el marco funcional que ha de guiar el diseño técnico. Su contenido se estructura de la siguiente manera:

Requisitos. Conjunto de necesidades y/o condiciones a satisfacer por las herramientas a desarrollar. Estos podrán ser:

- No funcionales: los que definen propiedades y restricciones del sistema. Son aquellos a los que debe apuntar la arquitectura puesto que si no se cumplen, el software puede no funcionar.
- Funcionales: aquellos que denotan la funcionalidad del sistema a desarrollar indicando de qué manera dicho sistema debe reaccionar a determinadas entradas, especificando cómo debe comportarse en situaciones particulares y declarando explícitamente lo que el sistema no debe hacer.

En la definición de los requisitos del sistema se establecen dos actores:

- Usuarios. Enfocado a quien vaya a utilizar la interfaz web para la visualización de la información.
- Sistema. Es la parte lógica del sistema, los componentes que se conectan a ella y sus comunicaciones.

### 4.2.1 Requisitos no funcionales

- RNF1. El sistema tiene que estar desarrollado con software libre. El software libre permite obtener soluciones que sigan estándares y a un coste nulo. La principal ventaja es la gran participación de la comunidad en la resolución de problemas y su fácil disponibilidad.
- RNF2. Uso de estándares abiertos. Esto permite que la comunicación entre los módulos del sistema sea ordenada y visible por cualquier otro sistema. Los estándares que más se adaptan al proyecto son los establecidos por el OGR y W3C.
- RNF3. Modelo de datos adaptativo a cualquier municipio. Diseño de datos que comunica a la interfaz web debe contemplar que hayan nuevas ciudades.
- RNF4. Datos fáciles de obtener y adaptar al sistema. La elección de las fuentes de datos se tiene que basar en un acceso sencillo a los repositorios y con datos en formatos estandarizados.
- RNF5. Los datos con información espacial. Los datos deben estar georeferenciados para poder realizar análisis geográficos.
- RNF6. Base de datos con tiempos de respuesta en consultas bajos.
- RNF7. Los datos a mostrar sean claros.
- RNF8. Interfaz web adaptada al usuario. El diseño tiene que contar con una buena experiencia del usuario y que sea adaptativa a cualquier dispositivo.
- RNF9. El sistema permitirá la inclusión de nuevos módulos en la capa lógica.

### 4.2.2 Requisitos funcionales

- RF1. La solución tiene que ser una aplicación web con diseño *responsive*.
- RF2. Datos abiertos portales de transparencia.
- RF3. Se establecen los barrios como unidad de medición.
- RF4. El usuario podrá conocer datos de un barrio seleccionándolo en un mapa.
- RF5. El usuario será capaz de comparar distintos barrios seleccionando en un mapa y que sean añadidas las métricas de forma dinámica.
- RF6. Realizar búsquedas por nombre de barrio.

- RF7. El usuario puede ver un ranking de los 10 mejores barrios de cada analítica.
- RF8. Representación de datos cartográficos con visores web.
- RF9. La aplicación tiene que contar con una gestión de usuarios.
- RF10. Registro de la actividad de usuarios.

### 4.3 Estudio de alternativas

En este apartado se expondrán las opciones posibles para el desarrollo del proyecto. Las alternativas tecnológicas que se adaptan al proyecto tienen que cumplir los requisitos especificados en el documento. En los siguientes apartados se determinarán las tecnologías usadas en el proyecto para poder realizar el diseño de arquitectura del sistema.



Figura 13: Alternativas tecnológicas. Fuentes [aws.com](http://aws.com), [mongodb.com](http://mongodb.com), [postgis.us](http://postgis.us), [python.org](http://python.org), [fullstackpython.com](http://fullstackpython.com), [leafletjs.com](http://leafletjs.com), [ubuntu.org](http://ubuntu.org)

#### 4.3.1 Base de datos

- MariaDB. Derivado de MySQL en su versión 5.1, es un gestor de base de datos relacional con un lenguaje SQL con licencia GPL. Permite la indexación de datos espaciales.
- MongoDB. Es un tipo de base de datos NoSQL de código libre, la diferencia con el clásico gestor de base de datos es el tipo de almacenamiento, puesto que almacena objetos BJSON un binario de JSON. El lenguaje de utilización es JavaScript, lo que supone una ventaja para la creación de aplicaciones web o móvil. Este sistema permite la indexación espacial de forma sencilla.

- PostgreSQL + Postgis. Postgres es un SGBD relacional SQL que junto a su extensión postgis los convierten en líderes en el almacenamiento y tratamiento de la información geográfica. Los casos de uso son innumerables, pero se resaltan las conexiones a SIG, a servidores de mapas OGR, drivers para gran parte de los lenguajes más utilizados. Por esta razón supone la mejor opción para el desarrollo en la capa de datos y completa los requisitos relacionados con la capa de datos.

### 4.3.2 Procesamiento de datos

- SQL. Structured Query Language, es el lenguaje de consultas para base de datos relacionales, software como Oracle, MySQL y PostgreSQL lo usan como interfaz de acceso a datos. Las principales funciones en el procesamiento de la información son la multiplicación de tablas o la agrupación. El conocimiento de este lenguaje es fundamental para poder realizar una buena administración de la base de datos, generar los diseños de modelo de datos, así como conocer las técnicas de consulta apropiadas para obtener un resultado eficiente. La ventaja es la posibilidad de realizar análisis espacial con PostGIS.
- Python. Es un lenguaje de programación multiparadigma e interpretado. La principal ventaja es la limpieza en el tipado del código y su sencillez. Este lenguaje tiene una gran popularidad en el ámbito de la información geoespacial, ya que diferentes SIG se han desarrollado con Python o incluye una API para el uso de sus funcionalidades como: QGIS (PyQGIS), GVsig (gvpy), Grass (pygrass) o ArcGIS (arcpy). Por otro lado, para el geoprocesamiento de datos en memoria se usa GDAL/OGR con una amplia funcionalidad tanto para datos vectoriales (lectura y escritura de formatos OGC) tanto como para ráster. Para el procesamiento y análisis complejos de datos, *pandas* es una librería que usa las estructuras de datos de numpy para la mejora de rendimiento con grandes volúmenes de información. También es posible realizar análisis de texto con la librería NLTK para Procesamiento de Lenguaje Natural y Scikity-Learn para procesamiento de datos y Machine Learning.
- R. Es el lenguaje de programación libre científico más usado en este momento. Está teniendo gran popularidad debido a su gran capacidad de cálculo y a librerías para el procesamiento de datos a partir de técnicas como Machine Learning. R también tiene una librería de procesamiento de datos espaciales (rgdal). Este lenguaje tiene gran versatilidad en el tratamiento de datos, pero no es comúnmente utilizado para el desarrollo de software.

### 4.3.3 Framework web

- Symfony. Está desarrollado en PHP cuyo patrón de desarrollo se basa en MVC es uno de los framework más utilizados en proyectos web ya que permite hacer desarrollos rápidos de aplicaciones. Con este framework se han desarrollado multitud de aplicaciones web como es el caso de Drupal, uno de los CMS libres más utilizados en la actualidad.

- Ruby on Rails. Es un MVC desarrollado en Ruby utilizado por empresas como Carto para el desarrollo de aplicaciones web.
- Django. Es un framework web MVC basado en Python muy consolidado y fuertemente testado. Este framework permite un desarrollo rápido y adaptativo API RESTful gracias a su extensión Django REST framework. Contempla el uso de URL personalizadas y adaptación de un sistema de usuarios. Este framework ha sido el utilizado en el presente proyecto al estar basado en Python, lenguaje muy utilizado para aplicaciones GIS.

#### 4.3.4 Web mapping

La visualización de datos forma parte de los requisitos marcados para este proyecto. Para la representación de datos geográficos en un entorno web, se necesita de un entorno que pueda mostrar datos en formatos propios de GIS adaptados a la visualización web (GeoJSON, WMS, WFS) para ubicar en el espacio datos vectoriales e imágenes. Las soluciones web para estas visualizaciones están desarrolladas en JavaScript y las más importantes son:

- OpenLayers. Librería JavaScript para visualización de información geográfica. Es la librería líder en funcionalidades GIS. Soporta todos los protocolos utilizados en GIS OGC para web.
- Leaflet. Librería JavaScript creada para un desarrollo sencillo y fácil. Esta librería es más reducida en su parte nativa que OpenLayers, pero dispone de funcionalidades desarrolladas por su comunidad en forma de plugins muy interesantes como heatmaps, clusters o control layers. Esta librería ha sido la elegida para el desarrollo de mapas web ya que permite interactuar fácilmente con el DOM y captar eventos necesarios para completar algunos de los requisitos.
- Google Maps API. La librería más utilizada en el desarrollo web debido a sus amplias funcionalidades que permiten realizar visualizaciones interactivas. También dispone de una documentación muy completa y una gran actividad en la resolución de problemas que ayudan al desarrollo. Pero el mayor atractivo es la integración con otras APIs de Google relacionadas con la cartografía, como es el geocoding o routing que localizan lugares en un mapa mediante direcciones postales.

#### 4.3.5 Gráficas

Otra de las opciones de visualización de datos para la aplicación, es mediante las gráficas ya que permiten analizar las distribuciones o proporciones de un conjunto de datos y facilitar la obtención de conclusiones.

- D3js. Es una librería JavaScript para la generación de gráficos e infografías mediante la creación de objetos SVG. Esta librería permite construir cualquier tipo de figura, pero tiene como desventaja el desarrollo que conlleva y la curva de aprendizaje.

- Chartist.js. Esta librería permite realizar los tres tipos de gráficas más comunes de forma sencilla y con un estilo limpio. La API está bien documentada con ejemplos sencillos para su implementación.
- Chart.js. Esta librería tiene como principal ventaja la generación fácil y con estilo de gráficas. Permite la creación de gráficas de barras, líneas, tarta, y polares. También permite la actualización de gráficas mediante JavaScript lo que supone acometer parte de las funcionalidades de los requisitos.

## 4.4 Diseño de solución tecnológica

### 4.4.1 Arquitectura del sistema

El diseño del sistema está basado en una arquitectura de capas típico de desarrollo de software. Se establecen tres capas: cliente, negocio y datos. La principal característica de este diseño es su aislamiento y definición de estándares de comunicación, de esta manera el sistema puede ser escalable y persistente. El sistema establece que las comunicaciones se basen en estándares y protocolos minimizando la adaptación de los distintos módulos si hay cambios en cualquiera de ellos.

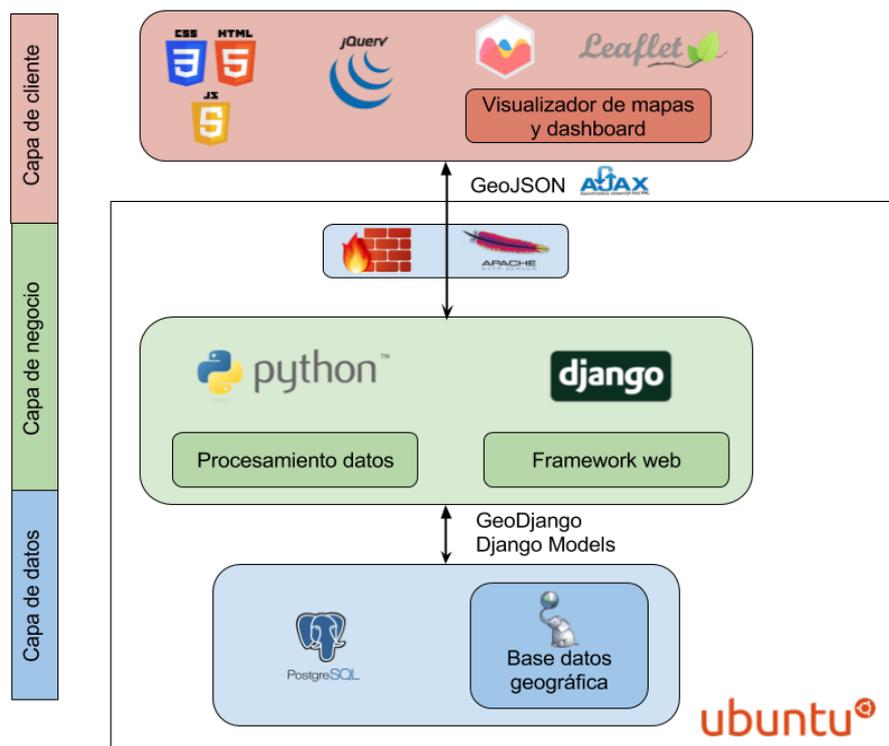


Figura 14: Arquitectura del sistema para la aplicación web. Elaboración propia.

Cada uno de los componentes que forman el sistema está diseñado para realizar funciones específicas que cubren las especificaciones requeridas para la solución planteada en el proyecto. También los módulos se definen para dar cabida a las distintas soluciones tecnológicas adecuadas para cada propósito garantizando los parámetros de usabilidad y desarrollo evolutivo eficiente (Figura 14). El sistema tiene que contemplar

que algunos de sus componentes puedan evolucionar dotándolos de nuevas funcionalidades adicionales a las expuestas en el proyecto.

Como entorno de desarrollo y de producción se establece el sistema operativo Ubuntu Server, éste y el resto de soluciones tecnológicas escogidas cumplen RNF1 de software libre. El servidor web que se encarga de las comunicaciones HTTP es Apache y CSF como firewall para configurar las conexiones seguras.

## 4.4.2 Capa datos

Para el almacenamiento y consulta de datos para el posterior análisis se usa como software PostgreSQL ya que dispone de las funcionalidades mínimas o exigidas para el diseño del sistema como es la capacidad de generar geometrías con el módulo PostGIS y poder realizar funciones de análisis espacial. Un ejemplo es conocer la distancia más grande a una calle (esto se puede extender a un conjunto de datos) o conocer las estaciones de Valenbisi que hay por barrio. Esta solución cubre los RNF5 y RNF6.

La capa de datos está compuesta por un módulo específico para la gestión de información geográfica, esto dota al sistema de capacidades para generar funciones de análisis espacial tales como la intersección, cálculo de distancias o corte de entidades. También, ofrece funciones de transformación de coordenadas, útil para pasar de coordenadas proyectadas (ETRS89) a geográficas (WGS84) necesario para la representación de los visores de mapas web según RF8. Gracias a estas funcionalidades espaciales se van a poder generar las estadísticas para las diferentes unidades que se establecen como objetivo del proyecto según RF3. Por tanto se pueden realizar procedimientos tanto en la parte del análisis como en la consulta de la información para la aplicación web.

Otras de las características de este sistema gestor de base de datos es la facilidad en la administración. Permite generar usuarios y roles de actuación. Gracias a esto, permite dotar de mayor seguridad al sistema a posibles ataques informáticos (inyección SQL). PostgreSQL permite la generación de esquemas en el modelo de tablas, esto permite realizar una abstracción de datos y agrupar tablas que puedan servir para unas funciones específicas del desarrollo o de procesamiento intermedio de aquellos datos que van a ser utilizados para la aplicación web, entre otras aplicaciones.

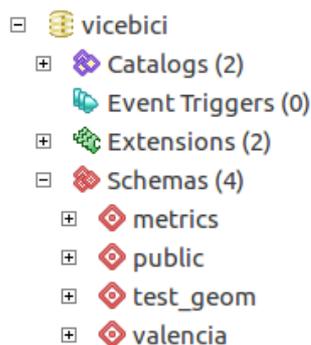


Figura 15: Distribución de esquemas para los datos del proyecto. Elaboración propia.

El modelo de datos es sencillo y no se establecen relaciones entre clases, lo que permite una fácil adaptación de cualquier ciudad que disponga los datos necesarios para la construcción de las métricas (apartado 5. Índices de movilidad urbana para bicicleta) RNF3.

### 4.4.3 Capa de negocio

En esta capa del sistema se desarrolla el componente lógico que realizan las funciones principales. En primer lugar se establecen dos módulos: framework y procesamiento de datos. Ambos están desarrollados en Python ya que permite generar un servicio web y generar algoritmos para el procesamiento y la carga de la información.

El framework tiene la función de gestionar la parte lógica de la aplicación web y los servicios de datos para la visualización en el cliente. Django es un framework del tipo Modelo Vista Controlador, de manera que mediante la configuración de la conexión a la base de datos, el modelo se encarga de adaptar los datos al framework. Las vistas se encargan de la generación de la interfaz de usuario con los datos que dispone. Por último el controlador se encarga de la gestión de los recursos y las peticiones de los clientes. Este patrón de diseño (también por capas) permite aislar los componentes, así como generar plantillas que dan la estructura de las interfaces.

Django también permite desarrollar un sistema de *login* que controla acceso a las partes de la aplicación que se establezcan como públicas. Esta funcionalidad relacionada con el RF9 y RF10, se basa en un sistema de registro de usuarios en la que mediante un correo electrónico y contraseña acceden a la parte privada de la aplicación.

El módulo de procesamiento se encarga de la carga de datos y de su transformación en el modelo adecuado. Esto se realiza mediante las librerías GDAL/OGR que permiten la lectura de datos de ficheros SHP o cualquier otro formato que cumpla los estándares OGC.

La comunicación con la capa de cliente se establece mediante mensajes en formato JSON y para los datos geográficos en formato GeoJSON. Este formato se adapta a los servicios web RESTful y el tratamiento de estos objetos es típico de aplicaciones web o móvil. El formato GeoJSON estandarizado nuevamente en

2016, tiene una estructura en la que se define un tipo, la geometría y las atributos del objeto. Este formato admite las siguientes geometrías:

- Point y MultiPoint
- LineString y MultiLineString
- Polygon y MultiPolygon

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [39.465, -0.37]
  },
  "properties": {
    "name": "Valencia"
  }
}
```

Figura 16: GeoJSON como formato para el servicio web de datos. Elaboración propia.

Este módulo también se conecta a la capa de datos mediante los *drivers* disponibles para Python como es *psycopg2*. Esto supone hacer una capa intermedia y de control de acceso a los datos abstrayendo cualquier actuación indebida. Para la configuración de dichas conexiones se establecen unas variables de entorno del sistema, de esta forma se extraen los datos de la aplicación web, ya que son críticos para el acceso a los datos.

#### 4.4.4 Capa de cliente

Esta capa se basa en un desarrollo web con los lenguajes que definen estas interfaces para navegadores, RNF2: HTML, CSS y JavaScript.

- HTML (HyperText Markup Language) definido por el estándar de la W3C, es un lenguaje de etiquetado que permite generar una composición, estructura y contenidos de una página web. La potencialidad de este lenguaje reside en la capacidad de estructurar y definir las partes de una página web, de esta forma se identifican los elementos y el tipo de estructura que se requiere de ellos. También define algunas tipologías multimedia que se quieren mostrar como imágenes, vídeos o enlaces.
- CSS (Cascading Style Sheets) es un lenguaje marcado que permite dotar a la estructura de contenidos y a los propios contenidos con estilos personalizados y algunos efectos básicos.
- JavaScript es un lenguaje orientado a objetos nacido para su aplicación directa en navegadores web. La W3C diseñó el estándar DOM para evitar incompatibilidades con su uso en distintos navegadores. Este lenguaje es utilizado principalmente para dotar de funcionalidades al DOM o generar procesos

en paralelo. Esto permite crear librerías o plugins que establecen funcionalidades específicas como generar canvases para hacer mapas o gráficas, también la modificación del propio DOM.

La interfaz gráfica web proporciona funcionalidades de utilización en diferentes dispositivos, es decir, es un entorno adaptativo o *responsive* RF1. Bootstrap es una librería Open Source compuesta por ficheros de estilos css y javascript que permite dotar de un estilo definido a una página web. Pero la principal ventaja del uso de esta librería reside en la forma de estructurar las páginas, se definen columnas y secciones con unas propiedades que determinan la manera en la que se va a visualizar dependiendo de la pantalla donde se cargue la página (Figura 17).

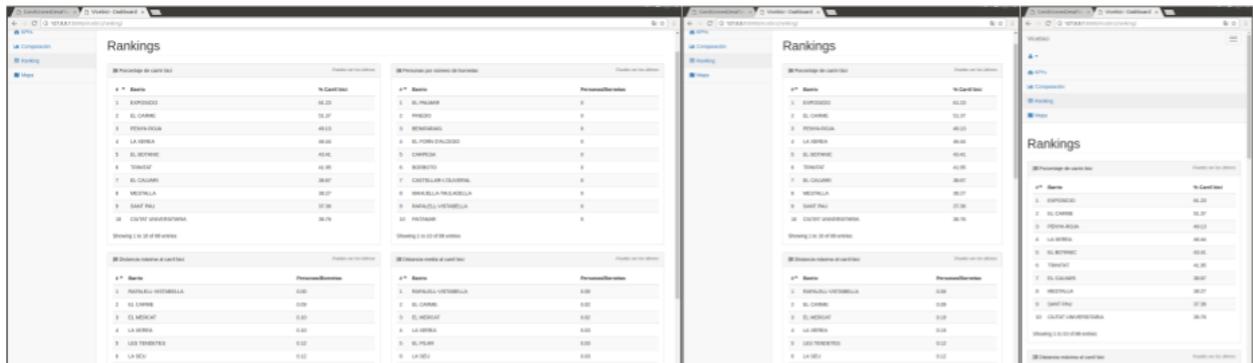


Figura 17: Muestra de ajuste responsive de la aplicación web con tres tipos de anchos de pantalla (1890px, 930px, 630px). Elaboración propia.

La interfaz web es donde se va a visualizar toda la información referente al objetivo del proyecto. Para este fin se han utilizado un conjunto de librerías javascript que permiten desarrollar diferentes sistemas de visualización y tratamiento de datos:

- jQuery
- Leaflet
- Chart.js
- jQuery UI
- DataTables

jQuery es una librería muy utilizada para el desarrollo web ya que tiene un conjunto de funcionalidades como inyectar HTML en un página web. Esto es necesario para dotar de dinamismo a los datos ya que permite interactuar con otros componentes de la página y quitar y añadir tablas o datos que se muestran en distintos elementos como proceso del navegador. Esta librería también permite realizar llamadas a servicios web de forma asíncrona, en el caso del presente proyecto es de utilidad para que la página se pueda cargar por completo y en un segundo plano obtener los datos que se quieren mostrar.

Leaflet es una de las librerías de referencia en *web mapping* y permite interactuar con elementos externos al propio mapa RF8. De esta forma se puede obtener información haciendo clic en un elemento del mapa y con

jQuery generar un HTML para inyectar en el DOM, RF4. Este es el caso de la información de las unidades de estudio que se muestran de forma dinámica en la página. También se puede mostrar esta información integrada en el visor mediante popups o con cuadros que se actualizan pasando el cursor por los elementos del mapa.

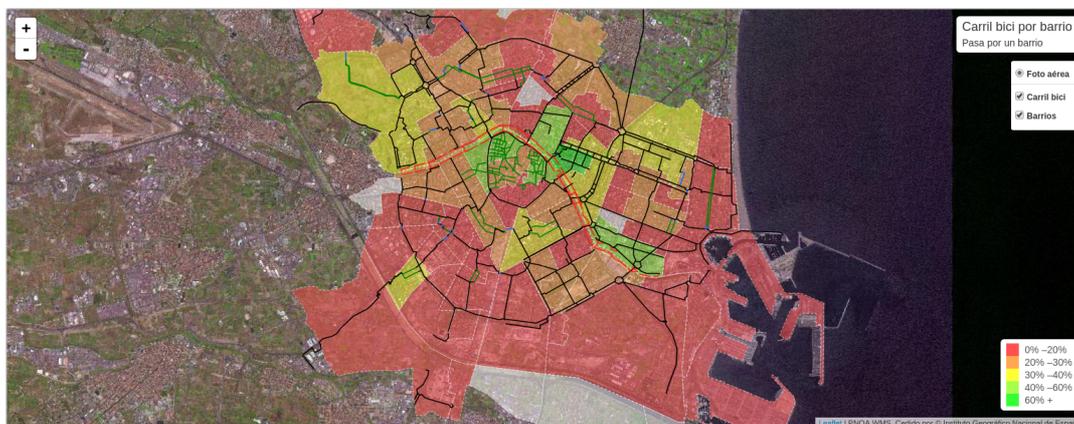


Figura 18: Mapa de coropletas del porcentaje de carril bici por barrio de Valencia. Elaboración propia.

También se pueden generar mapas de coropletas y leyendas con estilos personalizados y de forma dinámica, en el caso del proyecto se quieren mostrar mapas de forma que el usuario distinga las unidades de estudio con mejores índices de forma rápida y que se puedan identificar fácilmente (Figura 18). La información que se carga en estos visores se realiza mediante GeoJSON con los atributos que son establecidos en el framework web.

Otra de las funcionalidades de la página web es que se pueden mostrar los datos en gráficas de forma dinámica seleccionándolos en el mapa, RF5. Chart.js permite añadir datos y eliminarlos de forma dinámica sin tener que recargar la página web, de esta manera se han generado distintas gráficas que interactúan de forma dinámica junto con el mapa.

Para que el usuario pueda buscar un barrio, se habilita un elemento en la página que hace las funciones de buscador dinámico RF6. Para este fin se ha utilizado la librería jQuery UI, de forma que se puede hacer una conexión con el mapa y centrar la vista en el elemento que se ha buscado.

Otra forma de mostrar los datos es mediante tablas que indique un ranking de las unidades de estudio mejor posicionadas según los índices que se van a generar RF7. Para esto se utiliza la librería javascript DataTables que permite una ordenación de los datos dinámica y sencilla.

El desarrollo de las funcionalidades de esta capa será explicado más en detalle en el apartado de desarrollo del sistema junto con las acciones posibles que puede realizar el usuario de la aplicación.

## 5 Índices de movilidad urbana para bicicletas

Este apartado trata de explicar el diseño de uno de los objetivos principales, dar valor añadido a los datos que tengan que ver con la movilidad en bicicleta. Para ello en primer lugar se realiza un análisis de los factores más importantes que pueden afectar a esta materia y, en segundo lugar determina la metodología apropiada para el tratamiento de datos para su posterior visualización. En esta línea, se realiza un estudio de alternativas que puedan indicar el estado de la movilidad en la ciudad de Valencia.

En el presente estudio se va a hacer uso de diferentes fuentes de información para la obtención de conocimiento., para ello en primer lugar se estudia los datos relacionados con información sobre movilidad en Valencia.

### 5.1 Datos sobre movilidad urbana

Los datos que van a ser usados para el proyecto proceden del Portal de transparencia y datos abiertos (Ayuntamiento de Valencia) [8]. Este portal dispone de un catálogo de datos muy amplio y categorizado en distintas clases: medio ambiente, transporte, sociedad y bienestar, urbanismo e infraestructuras, salud y turismo. Los datos más representativos y afines al proyecto son:

- División administrativa de los distritos

Corresponde con las unidades administrativas de más alto nivel en el municipio, estos se dividen en los distintos barrios. Valencia cuenta con 22 distritos.

- División de barrios

Los barrios suponen la unidad más pequeña de división administrativa de la ciudad, cada uno de ellos está adscrito a un distrito. Valencia cuenta con 88 barrios. Ésta es considerada la unidad en la que se basarán los índices sobre movilidad.

- Carriles bici

Los carriles bici son las vías habilitadas para el tránsito de bicicletas. Estos datos cuentan con un atributo que define la tipología del carril: carril bici, Jardín del Turia, ciclocalle o vía peatonal. La tipología geométrica es de líneas.

- Aparcaderos de bici

Los aparcaderos de bicicletas que aparecen en estos datos hacen referencia a los situados en la vía pública. Estos son tipo punto.

- Estaciones de Valenbisi

Valenbisi es un servicio público de la ciudad de Valencia para alquilar bicicletas. Estos datos definen la posición de las estaciones que hay situadas en el municipio junto con el nombre, el identificador, el estado de

las bornetas, disponibilidad de huecos libres para aparcar, bicis disponibles y el total de bornetas que dispone la propia estación (Figura 19). Estos datos se actualizan con mucha frecuencia, de forma que pueden ser consultados por la API del ayuntamiento para ver la actividad del alquiler de las bicis en cada estación.



Figura 19: Estación de Valenbisi en avenida Tarongers compuesto por bornetas donde se estacionan las bicicletas. Fuente eldiario.es.

- Sentidos de circulación

Capa de puntos con los sentidos de circulación de las calles por las que transitan automóviles.

- Calles

Las vías en las que pueden circular cualquier vehículo se representan con líneas correspondiente al eje de la calle. De estos datos se puede obtener el nombre de la calle.

- Intensidad de tráfico de bicicletas

Medida de tráfico de bicicletas, capa de puntos con alta frecuencia de actualización. Esta información representa datos relacionados con la actividad en la circulación a partir de sensores (espiras electromagnéticas) que miden la densidad por hora del paso de vehículos.

- Secciones censales

Son divisiones territoriales en función de la población, estos se definen por una zona que contenga una población entre 500 y 2000 habitantes censados. Estos datos tienen una geometría de tipo polígono y son una división de los barrios de la ciudad.

- Manzanas con población

Las manzanas o islas catastrales, son agrupaciones de edificios rodeados por calles y vías (Figura 20). Estos forman polígonos que contienen la información agregada de población por cohortes de 0-14 años, 15-65 años y mayores de 65 años.

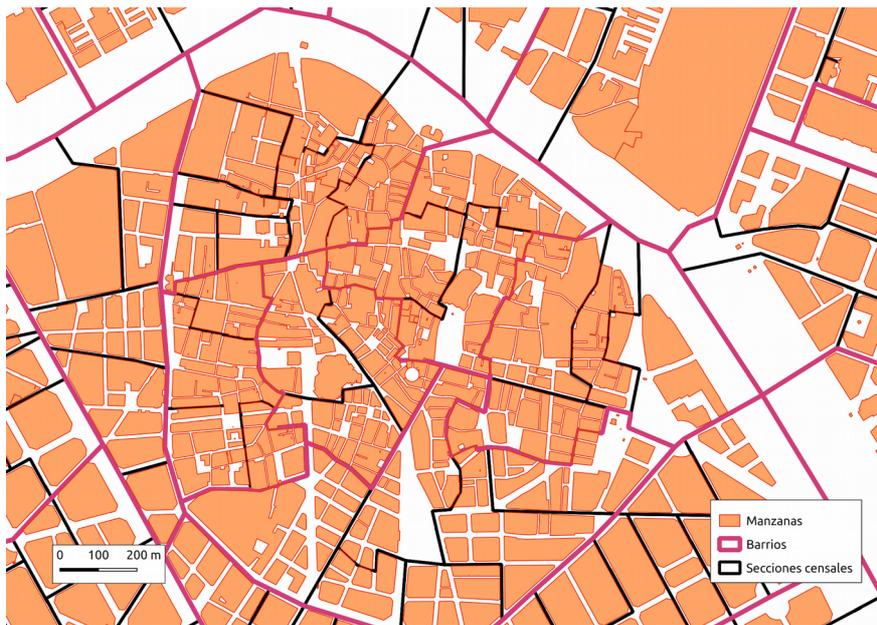


Figura 20: Representación de las islas catastrales, secciones censales y barrios en Valencia. Elaboración propia.

- Portales de fincas

Los portales son entidades puntuales que corresponden con los números de policía de los edificios.

Todas estas capas están disponibles con atributos espaciales en el sistema de referencia oficial ETRS89 en la proyección UTM huso 30.

## 5.2 Unidades de comparación

Para el análisis de los datos, se van a definir unas unidades que permitan hacer una agrupación de datos de forma que se pueda realizar estudios centrados en una región. Esto permite que toda la información espacial distribuida por el municipio se identifique con zonas determinadas para facilitar una comprensión mayor sobre la movilidad en bicicleta. Para realizar este estudio se cumple:

- Agregación simplificada de datos
- Análisis espacial y alfanumérico
- Relación de información de distinta naturaleza
- Modelización del estudio para aplicar en otras zonas geográficas

Se define como la unidad de medida el barrio. Este dato se puede obtener del Portal de datos abiertos de Valencia, de esta forma cumple con el requisito de poder ser aplicable el estudio en cualquier otra ciudad que disponga en su catálogo esta información. En caso de no disponer de esta distribución se puede optar por el uso de secciones censales disponibles en el INE junto a sus datos de población. La elección de esta distribución se basa en que todos los barrios pueden ser identificados de forma nominal haciendo posible que

una persona que resida en la ciudad pueda identificar con el nombre el barrio y de esta forma comprender mejor la división administrativa.

Para la relación de los datos espaciales con el barrio correspondiente se usan funciones típicas de los SIG. Las relaciones espaciales pueden ser implementadas de diferentes maneras, pero se usa una metodología que sea eficiente y con poco coste computacional, ya que el volumen de los datos puede ser muy grande y esto podría penalizar los tiempos de ejecución de las funciones espaciales.

Las relaciones entre los datos se realizan según una relación topológica espacial, éstas son definidas según la tipología de geometría que se quieren relacionar (define la dimensión y por tanto identifica el interior, borde y exterior de la geometría) y lo que se quiere extraer de esa relación (intersección, geometría que toca con otra, geometría que contiene otra...). Estas relaciones pueden ser definidas con la matriz DE-9IM (Dimensionally Extended nine-Intersection Model) o mediante funciones estandarizadas establecidas en PostGIS, evitando la elaboración de los predicados típicos de DE-9IM. La relación más utilizada en este caso es la intersección espacial de los datos con los barrios (Figura 21).

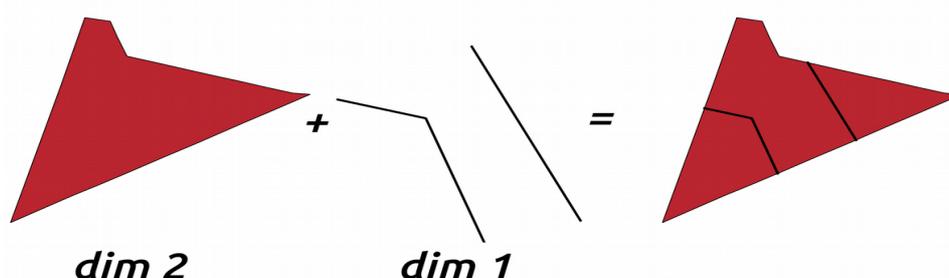


Figura 21: Intersección espacial de dos geometrías, la primera tipo polígono (dimensión 2) la segunda tipo línea (dimensión 1). Resultado de la intersección son líneas que intersectan con el polígono. Elaboración propia.

En el apartado sobre desarrollo se explica en detalle los datos que han sido relacionados espacialmente para la generación del análisis espacial.

## 5.3 Índices de estudio

Para realizar un análisis sobre la movilidad con los datos que se han presentado anteriormente, se plantea generar información derivada de los datos que proporciona el portal de datos abiertos para añadir valor y mayor conocimiento a los datos. De esta forma, el usuario de la aplicación, pueda entender e identificar de forma rápida zonas desfavorecidas o con ventajas sobre el resto.

Se han elaborado unos índices basados en datos relacionados con la movilidad como: carriles bici, calles, distribución de la población, estaciones de Valenbisi y portales. Con estos datos se han elaborado los siguientes índices:

### 5.3.1 Porcentaje de carril bici por vía urbana

Este índice consiste en una comparación de las infraestructuras destinadas a la circulación de automóviles y de bicicletas. Este índice se genera a partir de los datos espaciales (líneas) de forma que se extraen los metros de vía destinados en los distintos usos para obtener un porcentaje de metros que se destinan a la circulación en bicicleta. En este caso se puede conocer si el barrio que se analiza es bueno para el uso de bicicleta sin tener que usar la calzada destinada a los automóviles.

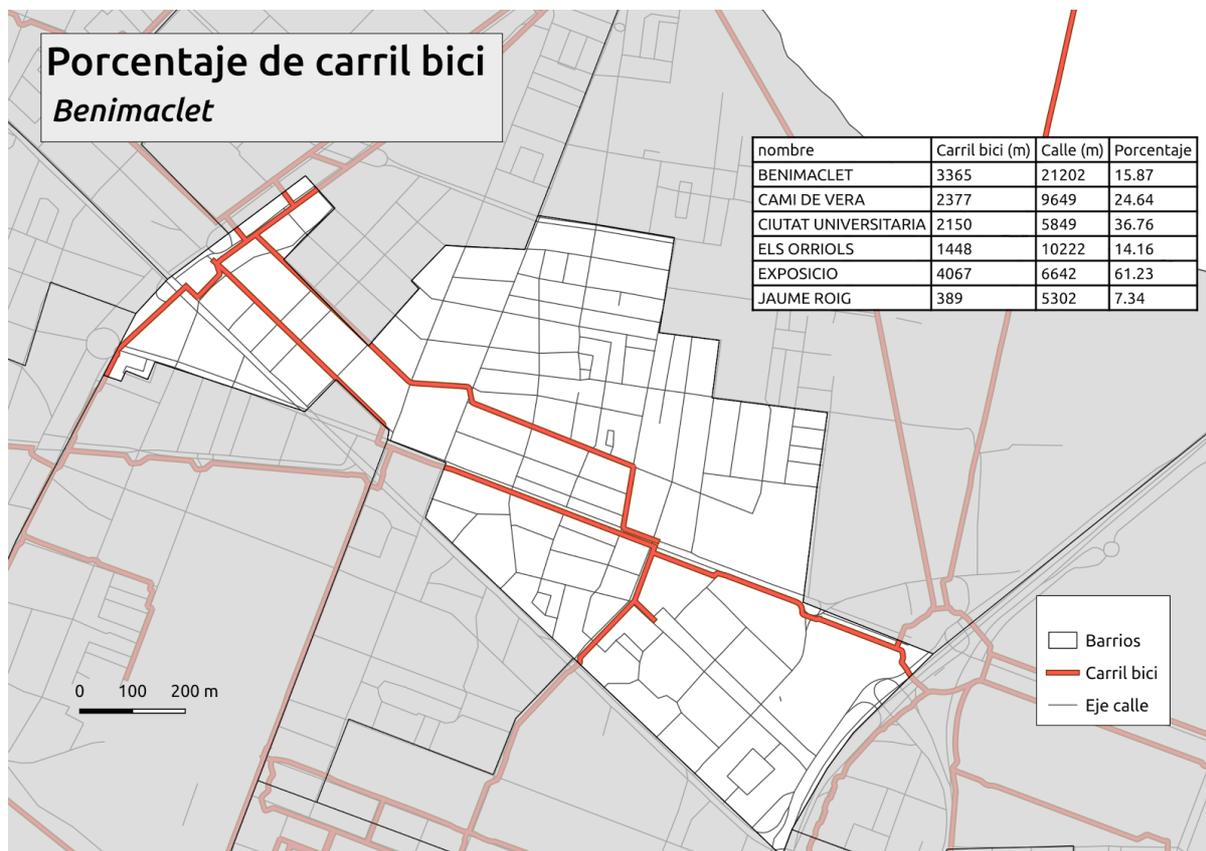


Figura 22: Cálculo del porcentaje de carril bici por barrio. Se muestra una tabla con los datos de los barrios contiguos a Benimaclet. Elaboración propia.

### 5.3.2 Población de un barrio por borneta de estación de Valenbisi

Este índice relaciona la distribución de la población por manzanas con las infraestructuras de Valenbisi que hay por cada barrio. En este caso se obtiene un coeficiente de población por las bornetas que existen en las estaciones de Valenbisi de un barrio.

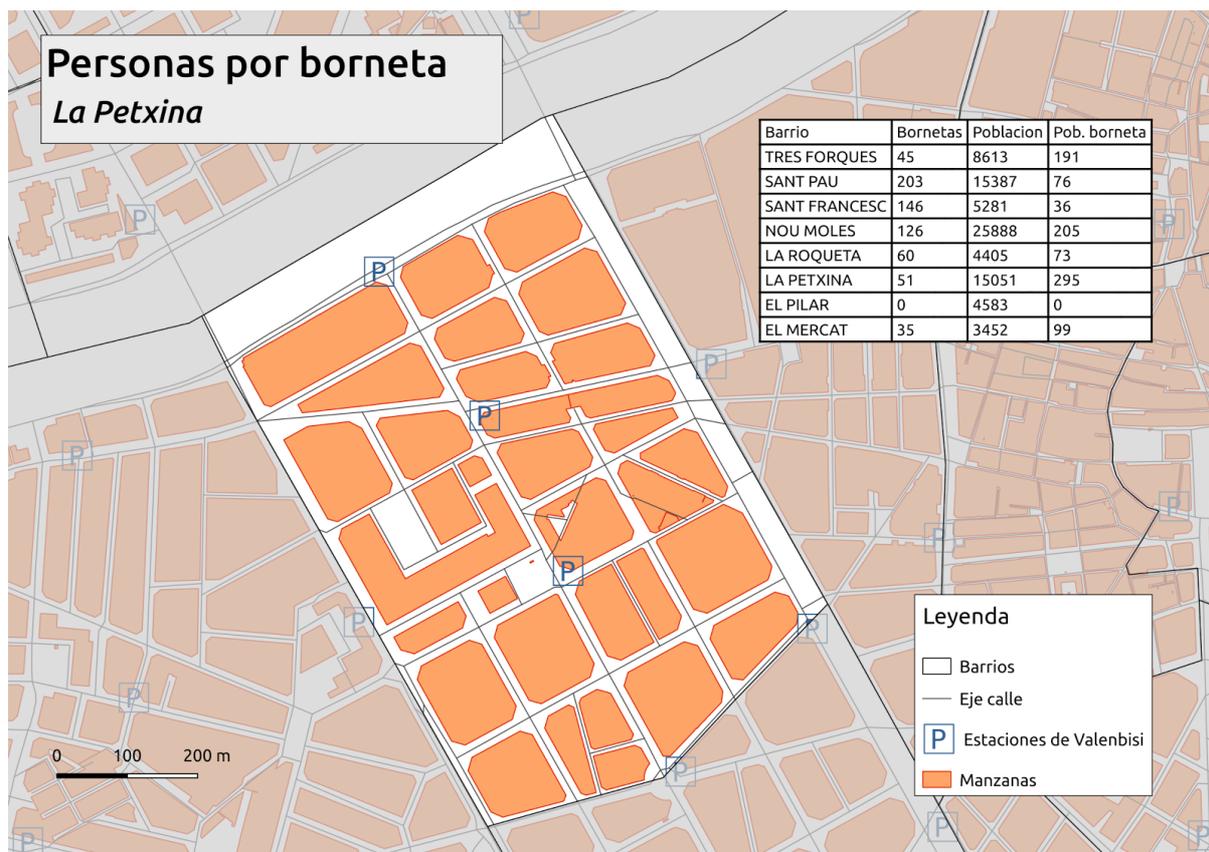


Figura 23: Índice de personas por borneta por barrio. Se muestra en el mapa el barrio de la petxina con los datos de manzanas y su población junto con las estaciones de Valenbisi. Elaboración propia.

### 5.3.3 Distancia media del carril bici a los portales

Este índice se calcula a partir de los datos de los portales (accesos al edificio residencial) junto con los carriles bici. Esto puede dar una idea de como es el acceso por parte de usuarios de bicicletas a las infraestructuras dedicadas a la circulación de las mismas. El caso claro es cuando una persona sale de su vivienda y tiene que recorrer una distancia hasta poder circular por un carril bici. Esto es importante ya que los ciclistas prefieren circular por estas vías ya que es más seguro y saludable.

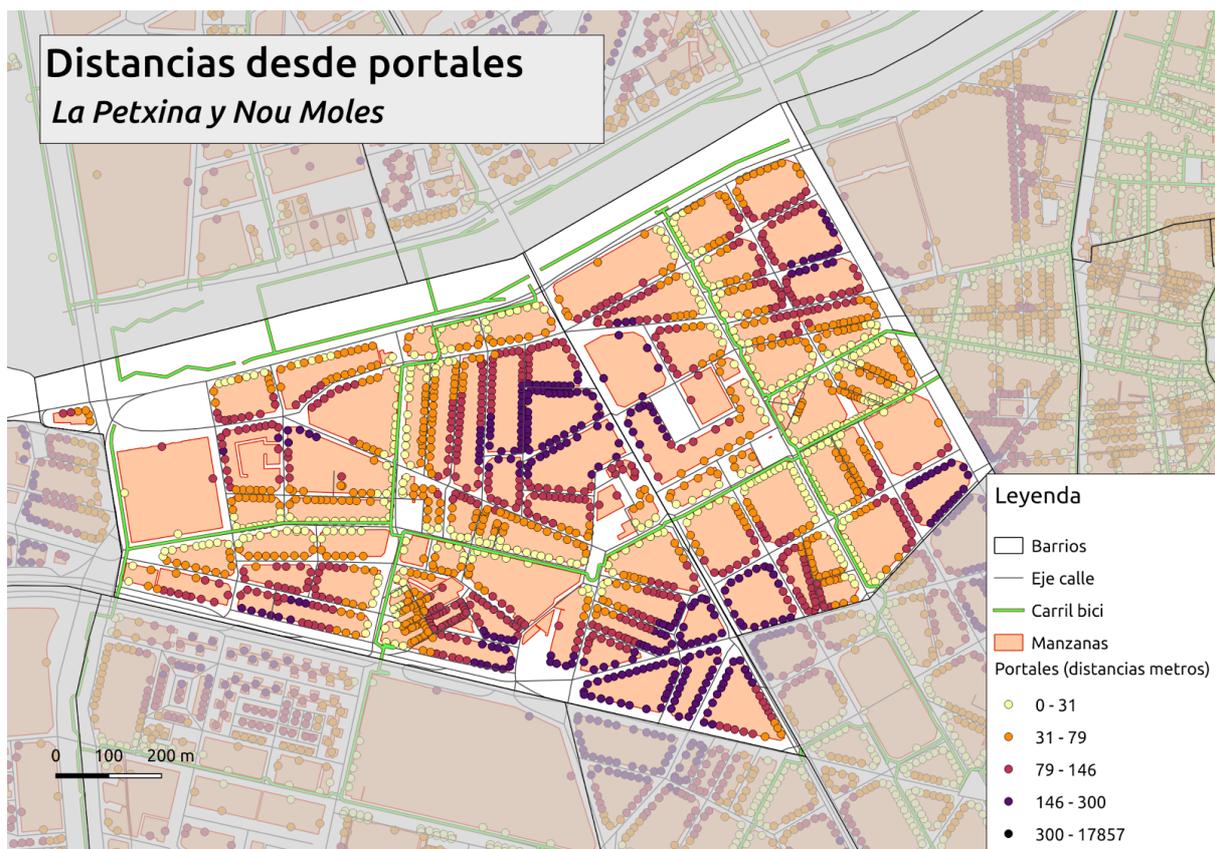


Figura 24: Representación de los portales con una rampa de color en relación a la distancia euclídea al carril bici más cercano. Elaboración propia.

En este índice se obtiene una media de las distancias calculadas.

### 5.3.4 Máxima distancia en un barrio entre portales y carriles bici

Este índice indica la distancia más grande entre un portal y el carril bici más cercano por barrio. Este dato señala el caso más desfavorable que existe entre el portal de un edificio y el carril bici más cercano, en términos de distancia euclídea (Figura 24). En este índice se puede ver como los casos más desfavorables son barrios situados en la periferia de la ciudad.

## 6 Desarrollo de la aplicación web

### 6.1 Entorno de desarrollo

El sistema definido anteriormente (Apartado 4. Diseño del sistema y componentes) se ha desarrollado en un sistema operativo Open Source Ubuntu 16.04, de esta forma se mantiene una coherencia entre el entorno de desarrollo y el que se usa para el despliegue de la aplicación de forma pública. El software necesario que compone la arquitectura del sistema es:

- PostgreSQL 9.5 junto con la extensión PostGIS 2.3.2.
- Python 2.7.12. Para el desarrollo de las diferentes funcionalidades se instalan los módulos:
  - Django 1.10.5
  - psycopg2 2.6.1
  - numpy 1.11

Para el desarrollo de la aplicación se ha hecho uso de los siguientes software libres:

- pgAdmin III. Cliente de escritorio para postgres que permite realizar un control de los datos almacenados de forma sencilla. Otra funcionalidad de la que dispone es la posibilidad de generar consultas para visualizar los datos. También permite conexiones a servidores alojados en otras máquinas, útil para el despliegue de la aplicación.
- Python virtualenvwrapper. Permite generar un entorno del intérprete python aislado con el principal del sistema, haciendo posible un mejor control del versionado de los módulos y la exportación de estos en otros entornos.
- Eclipse Neon. Software desarrollado en Java y que dispone del módulo PyDev para el desarrollo en Python. Se ha utilizado para el desarrollo del framework web, útil por la funcionalidad debug que permite realizar comprobaciones en el código. También dispone de un editor que ayuda al desarrollo ágil de software y crear estilos de visualización personalizados. Este software dispone de un módulo que permite crear y gestionar proyectos Django definiendo un entorno virtual con virtualenv.
- Sublime Text. Es un editor de textos desarrollado en C++ y Python al que se pueden adaptar diferentes sintaxis de lenguajes. Este editor es útil para el desarrollo de la capa de cliente, el cual se desarrollan las plantillas en HTML, los estilos CSS y las funcionalidades en JavaScript.
- Chrome. Este navegador basado en Chromium u otros como Mozilla, son necesarios para el desarrollo web y realizar los test necesarios de las páginas. también dispone de diferentes funcionalidades para el desarrollo de páginas web, como la inspección de elementos, los módulos y estilos.

- Git. Este software es utilizado para la gestión de versionado de los programas. Permite conectar a repositorios de forma que se garantice su persistencia y generar versiones de prueba, test y definitivas.

Todo este ecosistema para el desarrollo de la aplicación web es importante para asegurar un buen desarrollo del sistema. La configuración de cada uno de estos componentes ha sido minuciosa para su correcto funcionamiento.

### 6.1.1 Sistema Gestor de Base de Datos

Para el almacenamiento de la información se va a utilizar el SGBD PostgreSQL, de forma que será necesario instalar en el sistema dicho programa de forma correcta. También será necesario instalar la extensión PostGIS para el almacenamiento de datos espaciales. Para la gestión de la información almacenada existen varias alternativas, una de ellas pasa por el cliente en terminal que tiene PostgreSQL o psql en el cual se pueden realizar cualquier tipo de acción para la gestión de los datos (Figura 25). Esta alternativa es utilizada para labores de mantenimiento en servidores mediante conexión desde terminal con ssh, con la desventaja de que se tiene que conocer las funciones bien y dada la gran cantidad de éstas la curva de aprendizaje es muy pronunciada. Para realizar una gestión más ágil y visual de la base de datos se utiliza el cliente pgAdmin.

```
barroquia@barroquia-CX62-6QD:~/python/eclipse/cartos$ sudo -l -u postgres
[sudo] password for barroquia:
postgres@barroquia-CX62-6QD:~$ psql
psql (9.6.2, server 9.5.6)
Type "help" for help.

postgres=# \connect vicebici
psql (9.6.2, server 9.5.6)
You are now connected to database "vicebici" as user "postgres".
vicebici=# \dt *.*
vicebici=# \dt
          List of relations
Schema |          Name          | Type  | Owner
-----+-----+-----+-----
public | aparcabicis            | table | postgres
public | auth_group             | table | postgres
public | auth_group_permissions | table | postgres
public | auth_permission        | table | postgres
public | auth_user              | table | postgres
public | auth_user_groups       | table | postgres
public | auth_user_user_permissions | table | postgres
public | barrios                | table | postgres
public | bicicarriles           | table | postgres
public | distritos              | table | postgres
public | django_admin_log       | table | postgres
public | django_content_type    | table | postgres
public | django_migrations      | table | postgres
public | django_session         | table | postgres
public | eje_calles             | table | postgres
public | pob_manzanas           | table | postgres
public | portales               | table | postgres
public | spatial_ref_sys        | table | postgres
public | trafico_carriles       | table | postgres
public | valenbici_estaciones   | table | postgres
public | vicebici_ciudad        | table | postgres
(21 rows)
```

Figura 25: Cliente psql de PostgreSQL. Elaboración propia.

El framework web (capa de negocio) se va a conectar a este servicio para realizar las peticiones de datos, para ello será necesario crear un nuevo usuario para conectarse a la base de datos. A este usuario se le asignará unos roles específicos y distintos al llamado *superuser*, que es el que gestiona el correcto

funcionamiento de la base de datos y dispone de todas las posibles acciones a la base de datos. Los roles del nuevo usuario se asignan en función del alcance que tenga diseñada la aplicación que se quiera conectar. Es decir si la aplicación solo va a realizar consultas a un esquema o a una tabla en concreto, se pueden crear restricciones de tipo “lectura” para que sólo suceda este caso y no haga otras funciones CRUD (create, read, update y delete).

## 6.1.2 Framework web

La capa de negocio está desarrollada en base al lenguaje de programación Python y Django para el framework web de la aplicación. Para ello se ha instalado los diferentes componentes necesarios y para el desarrollo se ha utilizado un entorno virtual para el intérprete de Python llamado Virtualenvwrapper. Para ello es necesario instalar tanto virtualenv como virtualenvwrapper en el sistema y una configuración que pasa por definir donde se van a ubicar los entornos virtuales para Python y otras variables de entorno.

Eclipse es el IDE elegido para el desarrollo, este permite gestionar proyectos Python y en concreto de Django. En primer lugar se debe definir que el intérprete que se va utilizar va a ser el creado por virtualenv, de forma que todos los módulos deberán estar instalados en este entorno. Es importante no confundir el intérprete con el lenguaje, Python puede instalarse de diferentes formas y versiones en el sistema (Ubuntu cuenta en su distribución con Python 2 y Python 3). La diferencia reside en si se quiere tener un conjunto de módulos que van a ser utilizados sólo expresamente por la aplicación, de forma que con virtualenv se replica el intérprete en una carpeta del sistema y se almacenan los módulos requeridos de manera que se pueda exportar lo necesario.

Django tiene un procedimiento definido para la creación de un proyecto:

1. Creación de un proyecto Django. `django-admin startproject`.
2. Creación de una aplicación del proyecto. `python manage.py startapp vicebici`.
3. Creación del modelo de datos. `python manage.py makemigrations vicebici`.
4. Migración a la base de datos del proyecto. `python manage.py migrate`
5. Creación de un sistema de usuarios. `python manage.py createsuperuser`.

Todas estas acciones pueden llevarse a cabo desde Eclipse gracias a un plugin que adapta todas estas funciones al IDE. Cuando se crea un proyecto en Django es importante configurar el fichero `config` adecuadamente (esta parte se define en el apartado de desarrollo web). Un aspecto importante de esta metodología es la posibilidad de ejecutar un proceso desde eclipse lo cual permite hacer paradas en la ejecución y controlar el correcto funcionamiento del código.

Django está pensando para que haya una ejecución de un proyecto en modo debug, de forma que todos los errores que puedan haber a nivel interno del software es mostrado en el navegador y permite identificar los errores. Esto será modificado en el momento en que se despliegue la aplicación de forma pública. Otra

ventaja es que se puede comprobar el funcionamiento ya que se despliega un servicio (como un servidor) en el puerto que se desee.

### 6.1.3 Repositorio Bitbucket

Como se ha definido anteriormente. En este proyecto se está utilizando una gestión de versionado del software (en concreto del proyecto web) con Git. Para ello se necesita disponer un repositorio en el cual poder almacenar los cambios. En este contexto se ha barajado dos opciones: GitHub o Bitbucket. Los dos tienen la ventaja de ser gratuitos y compatibles con Git, pero en la parte gratuita de GitHub, los repositorios son abiertos y accesibles a todo el público. Al ser un proyecto en desarrollo se ha optado por una cuenta (gratuita) de Bitbucket (Atlassian), que permite crear repositorios y compartirlo con otros desarrolladores mediante invitaciones.



Figura 26: Repositorio del proyecto en Bitbucket. Fuente bitbucket.org.

Esta herramienta permite realizar acciones de subida del código generado en una rama del proyecto. Git compara la diferencias que han habido con la anterior versión y muestra los cambios realizados. Luego cuando se ha hecho las pruebas pertinentes se realiza un “merge” de la rama que estaba en desarrollo para pasar los cambios a la rama principal que será la que vaya a ser utilizada en producción. Esto supone que en el caso de que haya problemas o imprevistos se pueda deshacer los cambios y utilizar la versión estable anterior.

## 6.2 Gestión y tratamiento de la información

### 6.2.1 Procedimiento de obtención de conocimiento

Para realizar un análisis a partir de los datos de movilidad definidos es necesario establecer una metodología que justifique los resultados. La metodología a seguir consta de los siguientes pasos:

1. Preprocesamiento. Consiste en la preparación de los datos obtenidos del portal de datos abiertos. Principalmente trata de controlar que no existan omisiones en los datos y que mantengan una estructura de datos que faciliten el tratamiento. Este procedimiento se realiza en la fase de importación de los datos, en la cual se asegura que todas las entidades tienen la estructura necesaria.
2. Transformación. En este paso se realizan todos los procesos previos a la realización del análisis. Principalmente consta de la generación de nuevas variables mediante operaciones de normalización

y agregación. Esta parte se realiza con procesos que se encargan del cálculo de los índices de movilidad (Apartado 6.2.5) y se añade un procesamiento de normalización para la obtención de los resultados por cada uno de los barrios analizados.

3. Interpretación y evaluación. Se identifican los patrones mediante la visualización con gráficas y que son realmente interesantes, para poder obtener conclusiones basadas en las medidas tomadas.

## 6.2.2 Modelo de datos

Este apartado se explica la forma en la que se almacena la información y la estructura que tienen estos datos, esto permite organizar e identificar los recursos de información que se usan tanto para la aplicación web como en el propio análisis. Un modelo de datos sirve para definir y entender la estructura lógica de la información.

El modelo de datos está dividido en esquemas que corresponden a los datos descargados de una ciudad y el esquema que contiene la información a mostrar por la aplicación. Para la introducción de los datos en el sistema es necesario crear el esquema con el nombre de la ciudad, en este caso “valencia”, el cual contiene las tablas con información que será procesada para el estudio de movilidad en Valencia. Esto permite aislar y organizar la información puesto que si se quiere introducir datos de otra ciudad, es muy probable que no haya coincidencia con los modelos de datos ya que no hay un estándar para la estructura de tipo datos. De esta forma se asegura una persistencia en el modelo en caso de actualización de los datos.

Para realizar el estudio y procesamiento de los datos es necesario realizar un pretratamiento de los mismos y analizar la estructura de las distintas capas de información. Se definen los atributos necesarios para cada uno de los casos definidos en el Apartado 5.3, garantizando su existencia y, por tanto la posibilidad de generar los índices de movilidad urbana (Apartado 6.3):

- Barrios: población, nombre y geometría tipo polígono.
- Calles: geometría de tipo línea.
- Carriles bici: geometría tipo línea.
- Portales: geometría tipo punto.
- Bornetas o puntos de anclaje: número de bornetas, geometría tipo punto.

Descargados los datos del Portal de datos abiertos de Valencia [8] se realiza una extracción o transformación de los datos necesarios al modelo

El único modelo de datos que si permanece estable es el que recoge la información que va a ser mostrada por la aplicación web. Éste está compuesto de una tabla con los diferentes atributos de cada unidad de estudio en un esquema llamado “metrics”.

barrios_stats
id: integer NOT NULL, nombre: character varying(80) poblacion: double precision id_pob: integer bornetas: double precision pob_est: double precision metros_bici: double precision metros_calle: double precision por_calle_barrio: double precision max_dist_port: double precision avg_dist_port: double precision geom: geometry

Figura 27: Tabla de estadísticas de barrios. Elaboración propia.

Como se puede ver en la Figura 27 no existe ninguna relación estricta con otras tablas existentes en la base de datos, de esta forma se pretende garantizar la escalabilidad limitando las restricciones y establecer la fuente de datos original como recurso principal. Esto conlleva a que se deban realizar procedimientos de tratamiento de datos que garanticen la calidad y la integridad de los datos originales y adaptándolos al modelo que se necesita para el análisis. La clase que define el modelo de las métricas contempla el almacenamiento de las ciudades que se deseen adaptar al modelo de datos del análisis.

### 6.2.3 Importación de datos

La importación de datos al sistema es el primer proceso que se realiza y guarda relación con el tratamiento de los datos puesto que se van a transformar los formatos originales de descarga al modelo de tablas y esquemas de PostgreSQL.

Para realizar este proceso en primer lugar es necesario crear una base de datos con todas las funciones cargadas de la extensión PostGIS. También se crearán los diferentes esquemas necesarios para la estructura antes descrita.

Los datos obtenidos por el Portal de datos abiertos de Valencia han sido descargados y almacenados de forma local. Estos datos están disponibles en formato ESRI Shapefile (SHP) puesto que contienen información geográfica (puntos, líneas y polígonos). Este formato tiene la ventaja de estructurar los datos en formato dBASE (.dbf) y los índices espaciales que puedan contener los registros. Para realizar un correcto análisis se comprueba que todos los datos no contengan errores como la omisión de datos.

La forma en la que ha sido migrada la información a PostgreSQL ha sido desde el software QGIS, mediante el asistente de carga en base de datos (Figura 28). Esta herramienta es sencilla y es necesario que los datos que se desean cargar en la base de datos estén previamente cargados en el programa. También es necesario realizar la conexión a la base de datos que se quiere cargar, para ello QGIS cuenta con un asistente para comprobar la conexión.

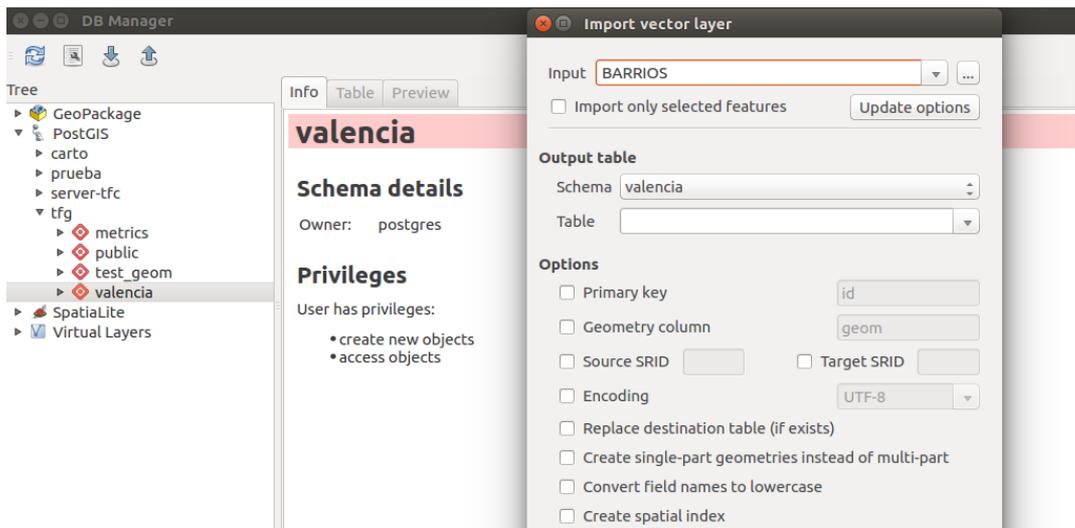


Figura 28: Migración de datos desde QGIS a PostgreSQL. Elaboración propia.

El asistente para la migración de los datos permite elegir el esquema y el nombre de la tabla que se va a generar en la base de datos de PostgreSQL. También se define tanto la clave primaria como el nombre de la columna que contendrá la información espacial. Las geometrías pueden ser migradas aplicando una transformación de sistema de referencia (definido en el fichero .prj), pero se va a mantener la información original y por tanto se asignará el sistema de referencia destino ETRS89 UTM huso 30 (EPSG:25830).

Otra opción para migrar los datos es mediante las funciones de postgres `shp2psql` en línea de comandos, esta opción permite la opción de automatizar el proceso de carga de los datos que son descargados mediante `scripts` en bash, esto no está contemplado en el alcance del proyecto pero supondría una opción de desarrollo en futuras líneas de mejora. Esta acción puede ser realizada mediante scripts en Python usando la librería GDAL/OGR que permite leer formatos SIG y conectar a base de datos para introducir la información.

## 6.2.4 Mantenimiento y procesamiento de datos

En este apartado se va explicar cómo se procesan los datos para obtener los índices de movilidad mediante un flujo de trabajo. Definida la estructura de la base de datos y la carga de información, se identifica y estudia la estructura de los datos que van a ser procesados para la obtención de los índices. Los algoritmos que se van a aplicar se realizan con funciones propias de PostGIS, de esta forma se generan unas vistas que contienen los diferentes algoritmos aplicados para la transformación de los datos.

Una vez generadas las vistas se realiza la migración de los resultados de las mismas a la tabla de estadísticas de barrios que contiene la información procesada. Este proceso se ha realizado con un script en Python de forma que realiza las inserciones de los datos de los barrios procesados. Este proceso se puede optimizar generando funciones que llamen al script mediante triggers, propios de PostgreSQL y se adopta como una de las líneas de mejora.

El desarrollo e implantación a otros datos no es objeto del alcance de este proyecto, pero si el diseño que permita la adaptación, la prueba ha sido realizada con los datos de la ciudad de Valencia pero el diseño se adapta a cualquier caso de estudio. El esquema de mantenimiento y generación de los datos se ha diseñado de la siguiente forma:

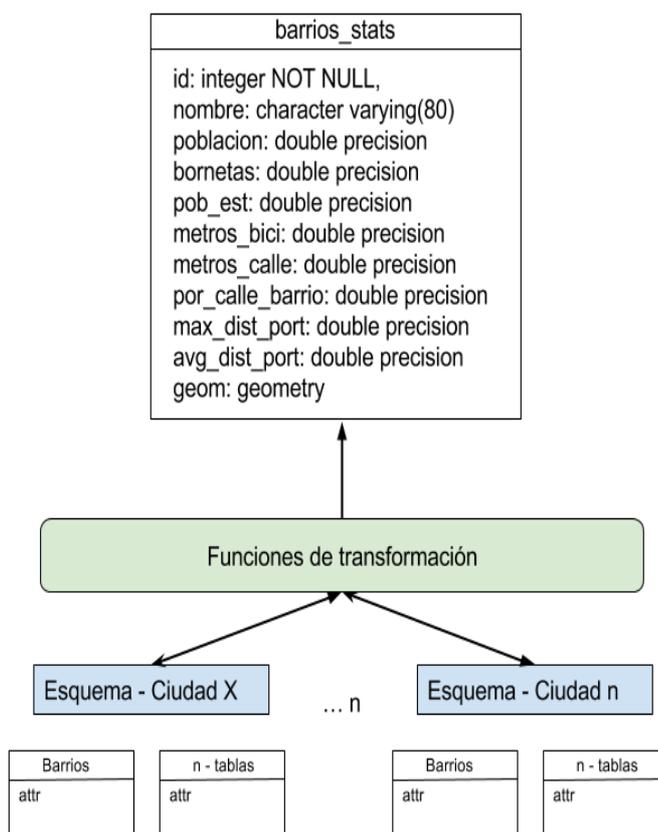


Figura 29: Modelo de generación de datos en el esquema metrics a partir de datos obtenidos de las distintas ciudades. Elaboración propia.

La aplicación se ha diseñado para el uso de datos de diferentes fuentes, es decir, los datos cargados en la primera versión, descritos anteriormente, procedentes del portal de datos abiertos de Valencia. En caso de que se quisieran añadir de otras ciudades y representarlo sería necesario crear un nuevo esquema en la base de datos e identificar los datos que guardan relación con los descritos para crear los índices.

### 6.2.5 Proceso de transformación de datos

Este apartado trata de explicar los algoritmos utilizados para la creación de los índices de movilidad urbanos así como la obtención de resultados del análisis. Todos ellos han sido generados con funciones PostGIS típicas de análisis espacial creando vistas de PostgreSQL mediante consultas SQL. Los resultados se han

generado mediante *scripts* en Python y volcados en la base de datos para su posterior publicación en la aplicación web.

El primer tratamiento realizado se basa en la obtención de los cuatro índices de movilidad urbanos:

- Porcentaje de carril bici



Figura 30: Aplicación de la función *ST\_Intersection* a la capa de carril bici con barrios. A la izquierda se aprecia como las líneas son continuas sin separación entre barrios, a la derecha se puede ver según el color el carril bici en función del barrio en el que se ubica. Elaboración propia.

Este índice se basa en una proporción de metros de calle con los metros de carril bici por barrio. Para la obtención de estas longitudes PostGIS dispone de una función que calcula la longitud de cada registro espacial. Para ello tanto las calles como los carriles bicis no tienen ninguna separación por barrio, es decir, una calle puede pasar por uno o varios barrios. Por esta razón no se puede realizar una simple relación espacial, es necesario obtener un recorte tanto de la capa de ejes de calle como la de carril bici con la capa de barrios. Esto es posible con la función *ST\_Intersection*.

```
-- partir carril bici por barrio
create view public.carrilbici_clip_barrio as
select id_b, clipped.barrio, id_c, clipped.estado, ST_SetSRID(clipped.clipped_geom,25830) as geom from
[
  (select pol.id as id_b, pol.nombre as barrio, ln.id as id_c, ln.estado,
  (ST_Dump(ST_Intersection(pol.geom, ln.geom))).geom As clipped_geom
  from public.barrios as pol left join public.bicicarriles as ln
  ON ST_Intersects(pol.geom, ln.geom)) as clipped
  WHERE ST_Dimension(clipped.clipped_geom) = 1;
]
-- agrupacion de tramos de bici por barrio
create view bici_barrio_grouped as
select max(id_b) as id, sum(ST_Length(geom)) as lon_bici
from carrilbici_clip_barrio group by id_b;
```

Figura 31: Sentencias *SQL* para división por barrio de los carriles bici y la agrupación por barrio. Elaboración propia.

Una vez cortadas las líneas, se realiza el cálculo de su longitud con *ST\_Length* y se hace un sumatorio por su relación espacial con los barrios. De esta forma se obtiene las agrupaciones tanto de metros de calle como de

carril bici por barrio. El cálculo del porcentaje se realiza considerando los metros de calle, el máximo rango de circulación y por tanto el 100% por lo que el carril bici es una proporción de este.

- Personas por borneta

Este índice expresa la relación directa entre las bornetas de las estaciones de Valenbisi de un barrio con el número de personas totales que residen en él. La capa de información donde se tiene la ubicación de las estaciones de Valenbisi dispone de la cantidad total de bornetas, junto a las manzanas con el dato poblacional, se realiza un sumatorio por barrio para obtener la cantidad total de ambos.

De esta forma se analiza la cantidad de personas por borneta que hay en un barrio. Para la obtención de este dato ha sido necesario realizar una relación espacial de los datos con los barrios con *ST\_Intersects*.

- Distancia media portales a carril bici más cercano

La distancia entre los portales y los carriles bici se obtiene con el cálculo de distancia entre geometrías mediante *ST\_Distance*. La Figura 32 muestra un ejemplo simplificado, en el cual un portal tipo punto tiene una distancia a la geometría tipo línea (esta será en el punto más cercano).

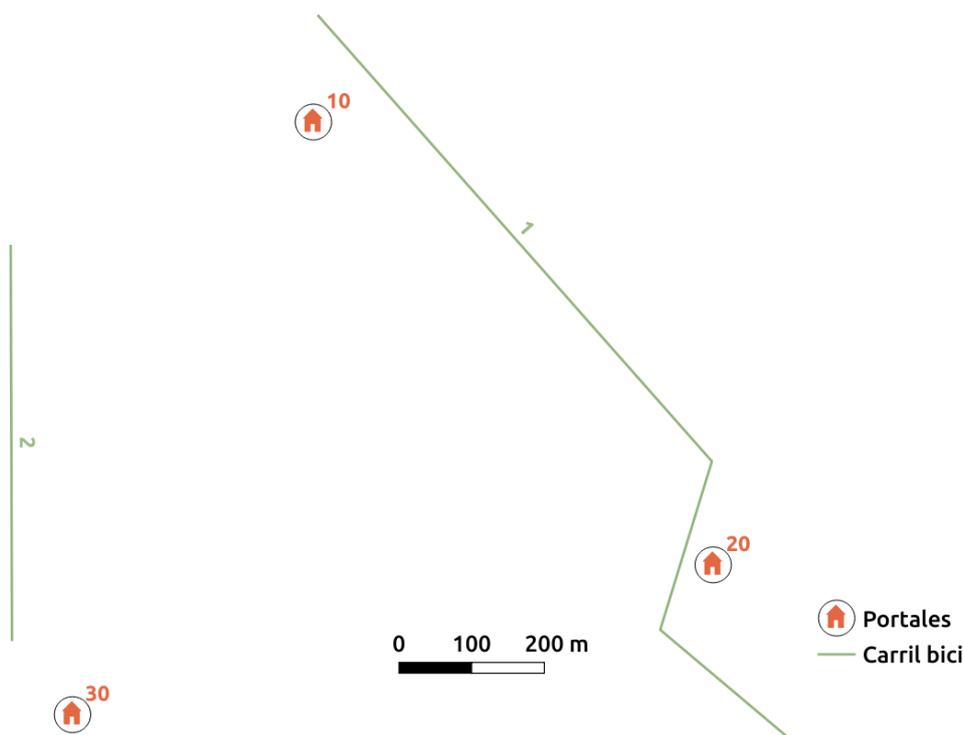


Figura 32: Ejemplo de distribución de los portales y un carril bici. Elaboración propia

Para conocer el carril bici más cercano es necesario realizar una multiplicación de tablas de portales con de la carril bici para obtener todas las posibles distancias entre todos los elementos espaciales.

	id_portal integer	id_carril integer	distance double precision
1	10	1	102.243787261842
2	10	2	450.309132843734
3	20	1	43.4954084785651
4	20	2	964.713792935299
5	30	1	816.843344151071
6	30	2	133.27563471414

Figura 33: Multiplicación de tablas con Left Join en PostgreSQL, lo cual relaciona cada portal con todos los carriles bici. Elaboración propia.

	id_portal integer	id_carril integer	distance double precision
1	10	1	102.243787261842
2	20	1	43.4954084785651
3	30	2	133.27563471414

Figura 34: Obtención de la distancia mínima de un portal al carril bici. Elaboración propia

Para obtener la relación de portal con menor distancia a un carril bici se ordena por id del portal y la distancia del portal, después se aplica la función distinct para obtener el primer valor que corresponde a la distancia mínima.

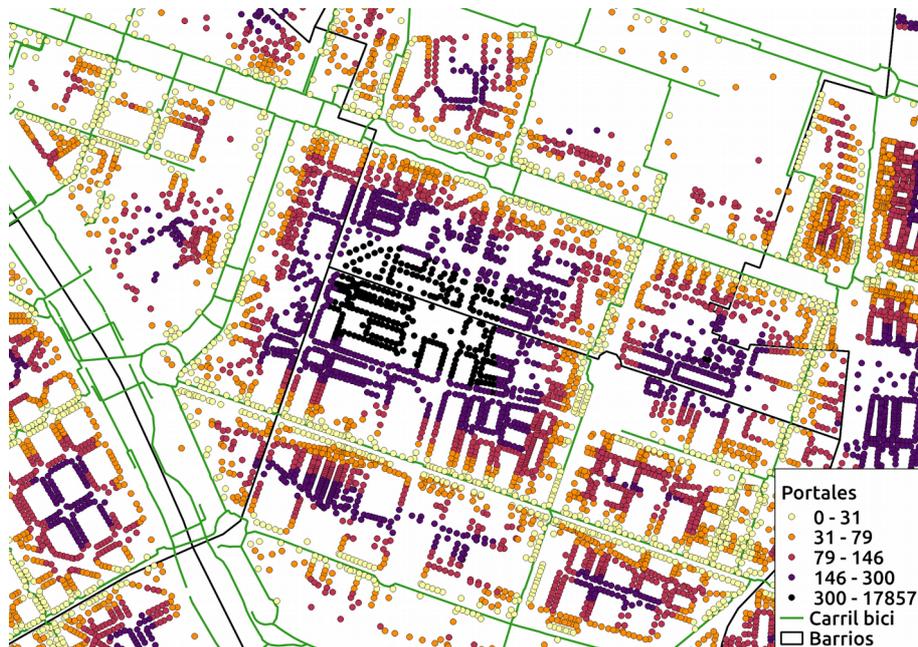


Figura 35: Mapa con los puntos representando los portales con una rampa de color en función de la distancia al carril bici más cercano. En el centro se observa una zona rodeada por carriles bici y como distancias desde el centro son mayores. Elaboración propia

Para la obtención de la distancia media es necesario que se haya asignado a los portales el id del barrio al que pertenecen mediante una relación espacial ST\_Intersects. De esta forma se tienen los portales con las distancias mínimas y mediante una agrupación se obtiene la media de los valores de distancia.

```
-- asignación de barrio a cada portal
CREATE VIEW public.portales_barrio AS
SELECT p.id,
       p.geom,
       b.id AS id_barrio,
       b.codbarrio,
       b.nombre
FROM portales p
LEFT JOIN barrios b ON st_intersects(p.geom, b.geom);

-- minima distancia entre portales y carriles bici
create view portales_mindist_carrilbici as
select DISTINCT ON (id_portal) id_portal, id_barrio, nombre, id_carril, distance, geom from
(select p.id id_portal, p.id_barrio, p.nombre, l.id as id_carril,
       ST_Distance(p.geom, l.geom) as distance, st_transform(p.geom, 4326) AS geom
from portales_barrio p , bicicarriles l ORDER BY p.id, distance) as dist;

-- distancias de portales agrupados por barrio
create view port_dist_barrio as select
id_barrio, max(nombre) as nombre_barrio, max(distance) as max_dist, avg(distance) as avg_dist
from portales_mindist_carrilbici group by id_barrio;
```

Figura 36: Sentencias SQL para el cálculo de distancia máxima y media por barrio mediante el uso de funciones de PostGIS. Elaboración propia.

- Distancia máxima portales a carril bici más cercano

Este índice se puede obtener con el mismo procedimiento que el cálculo de distancia media, de hecho se ha diseñado para que se comparta parte del algoritmo y sólo sea necesario obtener el valor máximo de distancia. Hay que matizar que el dato que se obtiene no es la distancia al carril bici más lejano, si no la distancia mínima de un portal al carril bici se obtiene el valor máximo de un barrio. Esto permite identificar los casos más desfavorables a la hora de acceder a un carril bici.

Este análisis y el anterior se basan en el cálculo de la distancia euclídea o el mínimo recorrido. Esto hace que el dato no exprese la distancia real de un portal al carril bici, pero da una idea aproximada de las magnitudes medias del barrio. De esta forma se abre otra línea de mejora en el desarrollo de un sistema basado en el análisis de topología de red o cálculo de rutas.

Tras el cálculo de los índices, estos reflejan unas características para cada uno de los barrios, para obtener un resultado de este análisis y evaluarlos de forma conjunta se realiza una normalización de estos índices para que puedan ser comparables entre ellos y se tenga una calificación de los barrios. Esta normalización se debe a que cada índice generado tiene unidades de medición distintas, procesando estos datos mediante un algoritmo de normalización como es el de mínimo y máximo se obtiene un valor contenido en un mismo dominio para todos, ejemplo entre 0 y 10.

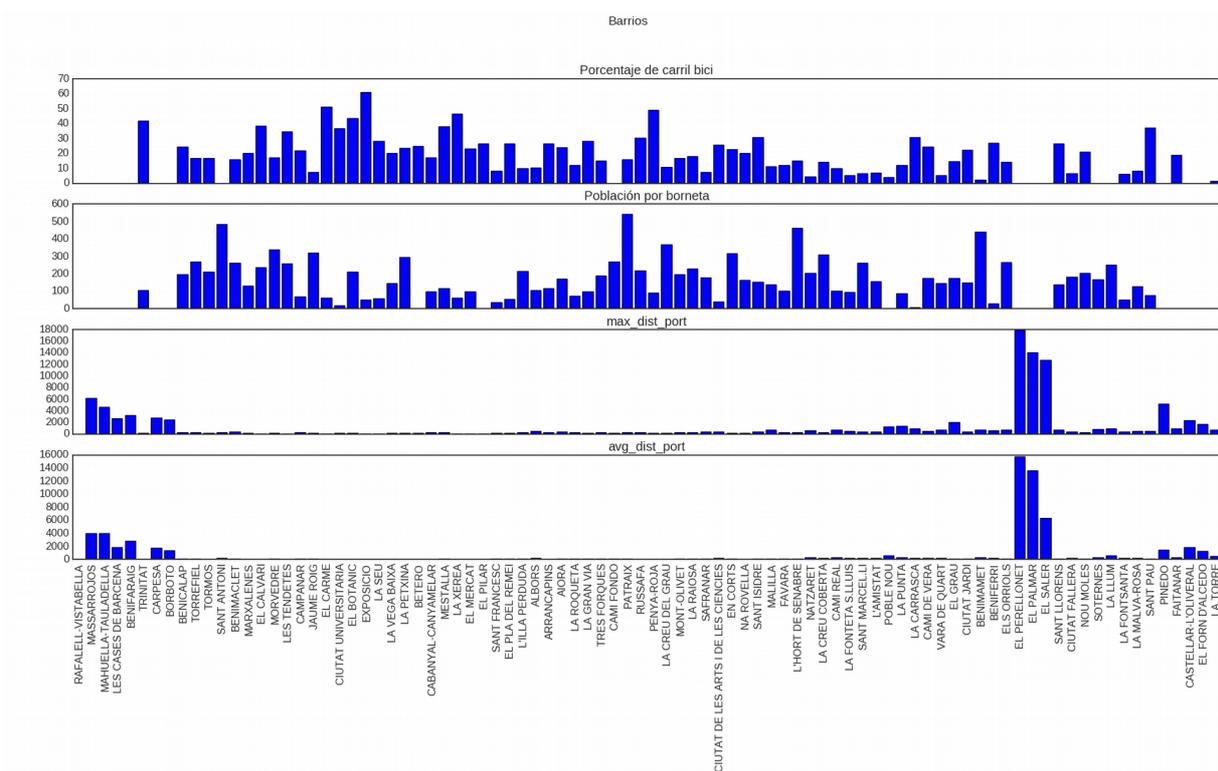


Figura 37: Gráfica de la distribución de los valores (eje de ordenadas) que tiene cada barrio (eje abscisa) por los cuatro índices calculados. Elaboración propia.

El primer paso para normalizar es conocer la distribución de la muestra en los 88 barrios que tiene Valencia para los índices calculados. En la gráfica (Figura 37) se muestran los índices (de arriba a bajo) de porcentaje de carril bici, personas por borneta, distancia media portales a carril bici más cercano y distancia máxima portales a carril bici más cercano.

El primer caso los valores fluctúan entre 100 y 0 dado que es un porcentaje se considera este índice como normalizado, el segundo los valores de población tienen una diferencia entre máximos y mínimos poco relevante según muestra la siguiente tabla (Figura 38). Sin embargo en el caso de los dos índices de distancia la diferencia si es sustancial, ya que se trata de índices relacionados con barrios situados en la periferia (El Saler, El Palmar o El Perellonet) y son los más desfavorecidos ya que la distancia de un portal al carril bici son muy grandes. y puede afectar al proceso de normalización según la distribución que tengan los datos.

	Porcentaje carril bici	Población por estación	Distancia máxima	Distancia media
<b>count</b>	88,00	88,00	88,00	88,00
<b>mean</b>	16,78	141,76	1224,22	768,73
<b>std</b>	14,08	124,51	2830,16	2354,27
<b>min</b>	0,00	0,00	0,00	0,00
<b>max</b>	61,23	543,91	17856,53	15744,74

Figura 38: Tabla de estadísticas de los índices calculados. Elaboración propia

Para el análisis de la distribución en el caso de los índices de distancias a portales más lejanos se hace uso de un histograma (Figura 37) para ver si contienen datos que puede afectar a la normalización. Este histograma refleja la acumulación o número de barrios con valores comprendidos intervalos iguales, en el caso de la figura de abajo son diez intervalos con la misma separación.

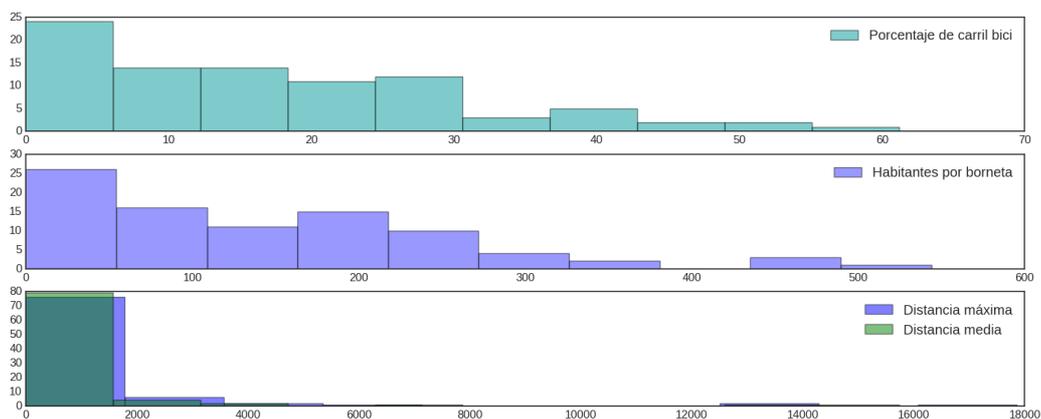


Figura 39: Histograma con valores acumulados en 10 conjuntos (bins). Elaboración propia.

Para que exista una distribución de los valores más homogéneos, como en las gráficas a y b de la Figura 39 se extraen los *outliers* asignando un valor que se considera frontera o límite de la normalización. Este valor frontera serán los casos con mayor distancia y tras la normalización estos barrios tendrán una calificación de barrio mínima, ya que tener mayor distancia supone una desventaja para los ciudadanos.

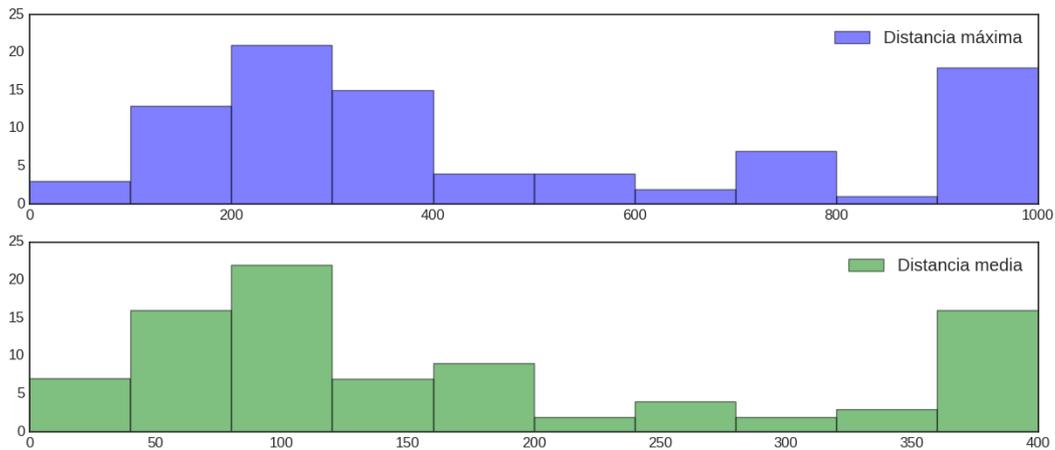


Figura 40: Histograma con valores acumulados en 10 conjuntos (bins) corrigiendo los outliers. Elaboración propia.

De esta forma se puede ver como queda la distribución de la normalización asignando como valores frontera 1000 metros en el caso de las distancias máximas y 400 metros en las distancias medias.

$$v' = \frac{v - v_{min}}{max_A - min_A} (newMax_A - newMin_A) + newMin_A$$

Figura 41: Fórmula de normalización máximo mínimo.

El proceso de normalización se realiza con el mínimo y máximo, y estableciendo el rango entre 0 y 10. De este cálculo se obtiene la valoración de cada barrio en los distintos índices siendo 10 la mejor valoración y 0 la menos favorable. Para los valores de porcentaje de carril bici cuanto más carril bici haya mejor será la valoración. Sin embargo en los casos de población por borneta, distancias máxima y media sucede al contrario, estos casos se invierte la calificación en la normalización.

Una vez normalizados se obtienen los resultados en los cuales se puede conocer el listado de los barrios ordenados con una calificación mediante la suma directa de los datos normalizados y dividiendo la por el número de índices (9.1 Tabla de resultados).

## 6.2.6 Resultados del análisis

Los resultados del análisis realizado en el apartado anterior consiste en una calificación de los barrios de la ciudad de Valencia en cuestión de movilidad urbana en bicicleta. Las calificaciones por barrio tienen un rango de 10 a 0, siendo 10 la mejor calificación posible y 0 la peor.

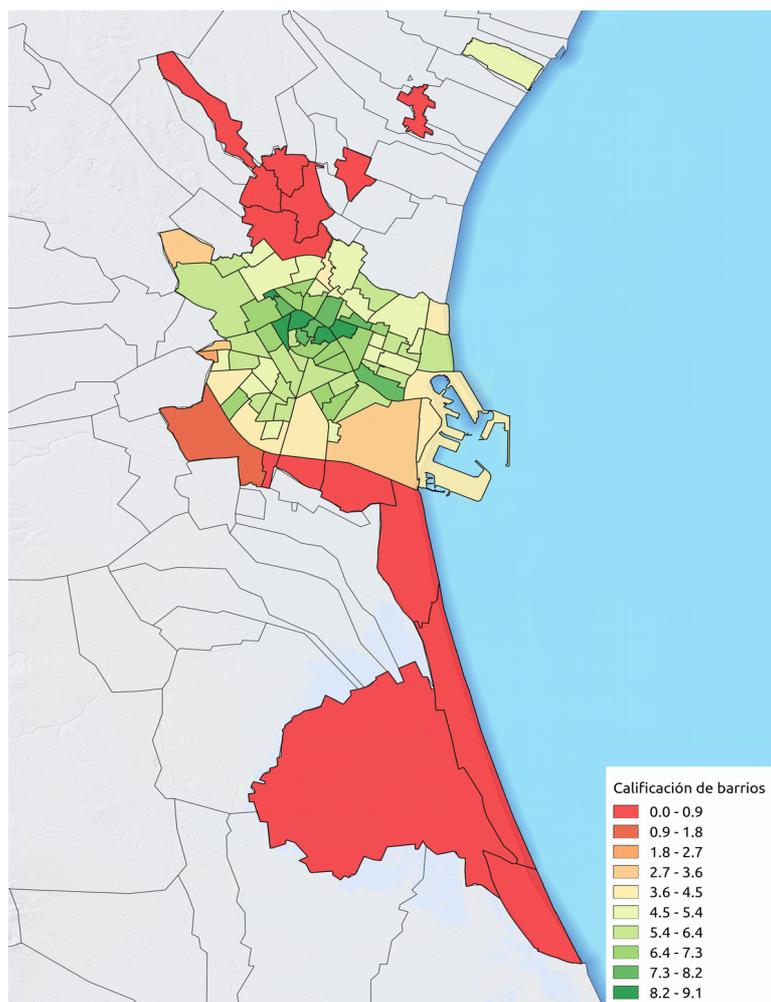


Figura 42: Mapa de coropletas de los barrios de Valencia con las calificaciones del análisis de movilidad urbana. Elaboración propia.

De los resultados obtenidos y partiendo de la Figura 42, se observa que:

- El barrio con mejor calificación es Exposició seguido de El Carme y La Xerea, barrios con buena distribución de carriles bici y estaciones de Valenbisi. Los barrios de los distritos de Poblats del Sud y Poblats del Nord son los que peores calificaciones obtienen ya que la red de carril bici es muy escasa y tiene ausencia de estaciones de estaciones de Valenbisi.
- Los barrios más céntricos y más pequeños son los que mejores puntuaciones tienen, esto se debe sobre todo a que la red de carriles bici que hay en estas zonas es mucho mayor y supone un mayor porcentaje en carril bici y distancias de portales a los carriles más pequeñas.
- Los barrios periféricos tienen baja calificación ya que en la cartografía no hay carriles bicis o su existencia es muy baja con respecto a las calles disponibles para vehículos. Esto hace que las distancias de portales a carril bici se incrementen, suponiendo un factor crítico que da a entender que estos barrios no son buenos para la circulación en bicicleta.

- La variable que más puede condicionar en la valoración es la distribución de los carriles bici, su existencia en un barrio puede suponer grandes variaciones en la calificación final. Este caso es notorio en barrios como Malilla (1,8 en nota de porcentaje carril bici) o El Cabañal (2,6 de nota) que tienen gran cantidad de calles y casi en su totalidad no están adaptadas a la circulación en bici, haciendo que su nota disminuya. En el caso de el Carmen (8,4 de nota) es lo contrario ya que la mayoría al ser peatonales o ciclocalles dispone también de una buena calificación global (8,8).
- En cuanto a los índices de distancia a carril bici más cercano, muestra la conectividad entre las viviendas con las infraestructuras de circulación como es el carril bici. La diferencia entre estos índices puede mostrar el grado de dispersión que hay en los barrios y destaca aquellos en que tiene zonas con mala conectividad aunque la distancia media pueda indicar lo contrario. Los casos de El Saler, El Perellonet o incluso dentro del núcleo de Valencia (El Grao) son los más desfavorecidos.

## 6.3 Desarrollo web

### 6.3.1 Diseño de aplicación web

El proyecto web se ha basado en una aplicación cliente/servidor en el cual el cliente realiza una petición que es enviada al servidor mediante TCP/IP. Este recibe la solicitud del cliente y realiza unos procesos para devolver una respuesta al cliente. Dentro de la tipología de servidores el utilizado para la aplicación de movilidad urbana en bicicleta es Apache ya que es un servidor que se comunica mediante protocolo HTTP con el cliente. Este servidor permite el uso del módulo WSGI para la conexión a aplicaciones web como Django.

Django es un framework web que permite gestionar las peticiones y dar respuesta a ellas, de forma que se puede generar procesos que permitan enviar al cliente ficheros (HTML, imágenes, JavaScript o ficheros de estilos css). Este desarrollo, correspondiente a la capa de negocio del proyecto, se denomina habitualmente desarrollo en servidor o *backend*.

Por otro lado, en la aplicación se necesita desarrollar la parte que será visible y funcional para el cliente o la interfaz web de la aplicación. Es por eso que se han diseñado unas plantillas HTML que junto con los estilos en css y funcionalidades en JavaScript permitirá al usuario visitar la aplicación desde un navegador web.

### 6.3.2 Desarrollo servidor

Esta parte del desarrollo es la correspondiente a los procesos y servicios que tienen que ver con la aplicación web, en esta parte se van a desarrollar todos los módulos o componentes que lo forman.

Django es el framework utilizado para gestionar tanto la lógica de la aplicación como los servicios que la componen. En primer lugar se ha creado un proyecto Django y una aplicación (llamada vicebici). Las aplicaciones dentro de Django son una forma de separar distintas funcionalidades o servicios que se quieran

hacer en un servidor. En el primer nivel de estructura se encuentra la carpeta correspondiente al nombre del proyecto en la que se destacan:

- `settings`: Es el fichero de configuración general del proyecto. En este módulo se definen los parámetros de conexión a la base de datos, la ubicación del sistema de ficheros estáticos, ubicación de las plantillas HTML y las aplicaciones que puedan ser visibles.
- `url`: En este módulo se definen las URLs que servirán de base para el proyecto. Cuando se genera una aplicación es necesario que se defina el patrón de URL por el cual se quiere acceder al subdominio.
- `wsgi`. El servidor apache recibe las conexiones del cliente y son transformadas en WSGI para que pueda ser interpretado por la aplicación de Django. Para eso es necesario configurar Apache y activar el “`mod_wsgi`” respetando las directivas de la configuración de Apache.

Una de las principales ventajas que tiene Django es su propio sistema de ficheros, este se genera en el momento de creación de una aplicación. El framework gestiona diferentes módulos para un correcto comportamiento de una aplicación interna de Django como:

- `migrations`. Esta carpeta contendrá las migraciones auto generadas por Django cuando se define alguna clase en el módulo `models.py`. Estos permiten la adaptación de las clases al formato de la base de datos (tabla).
- `static`. Es la carpeta que contiene todos los ficheros que van a ser lanzados a la aplicación ya sean multimedia o ficheros JavaScript o css. El uso correcto de esta carpeta es fundamental para una buena estructura y organización de los archivos, y porque Django permite una referencia en las llamadas a estos ficheros desde las plantillas. En esta carpeta es donde se alojan todos los plugins y librerías JavaScript y css.
- `Templates`. Esta carpeta contiene todos los ficheros HTML que se usan en la aplicación.
- `admin.py`. Módulo que contiene las funcionalidades internas del framework para la administración de contenidos y usuarios de esta aplicación.
- `models.py`. Es donde se generan las clases que referencian a las tablas. Esto es una de las ventajas que proporciona Django, permite generar el modelo de datos de forma que pueda ser exportable a cualquier base de datos compatible. También supone una ventaja ya que permite que se utilice el sistema ORM (Oriented Relational Model) propio de Django, esto permite abstraer el lenguaje de consulta propio de la base de datos ya que se usa el lenguaje orientado a objetos. Una vez que se definan las clases es necesario realizar el proceso de migración (`makemigrations` y `migrate`).
- `urls.py`. Este módulo es donde se definen las rutas de la aplicación, su relación con las vistas y el nombre. La definición de estos parámetros son muy importantes ya que se definen los parámetros de entrada a las vistas por medio de url y se pueden hacer filtros con expresiones regulares para definirlo.

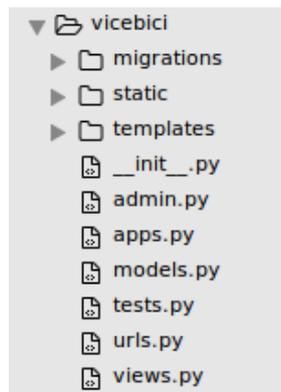


Figura 43: Estructura de ficheros de una aplicación Django. Elaboración propia.

En este proyecto Django se ha trabajado directamente en los ficheros contenidos en la aplicación vicebici, no se ha requerido de otro servicio o aplicación para el funcionamiento de la web. Como diseño de navegación se han definido cuatro URLs principales por las cuales el usuario tendrá capacidades de navegar, por otro lado existen otra serie de subdominios que consisten en la administración del proyecto, pero no es objeto del presente proyecto fin de carrera. Extraído directamente del código del módulo *urls* se definen:

- Index: `url(r'^$', views.index, name='index')`,
- Mapas: `url(r'^map/$', views.map, name='map')`,
- Comparación: `url(r'^comparation/$', views.comparation, name='comparation')`,
- Ranking: `url(r'^ranking/$', views.ranking, name='ranking')`,
- Datos: `url(r'^data/$', views.data, name='data')`,

Cada una de las URL han sido definidas con un patrón de expresiones regulares, en estos casos se define que el usuario, si quiere acceder a estos subdominios, tiene que poner exactamente lo que contenga entre “^” y “\$”. También se han definido las correspondientes vistas que gestionan las peticiones de la correspondiente url, estas vistas están definidas previamente en el módulo “views”. Por último se ha hecho uso de descriptores con el parámetro “name” para la definición de los mismos en las plantillas.

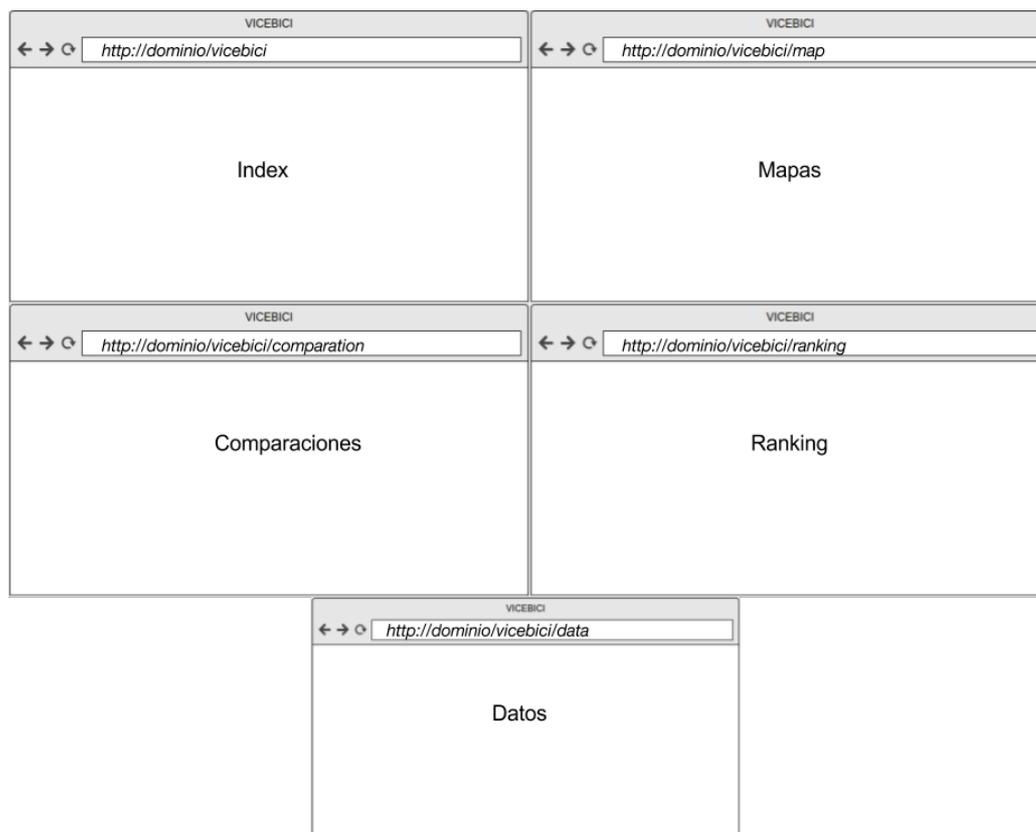


Figura 44: Diseño de mockups de las páginas con su correspondiente dirección URL y lo que se va a mostrar. Elaboración propia.

Una vez definida la estructura general, se desarrollan las vistas para relacionar las direcciones definidas anteriormente con las “plantillas” HTML que formarán el contenido de cada una de ellas. Las vistas creadas en el módulo *views* son funciones de Python en las que se definen las acciones que darán como resultado una respuesta al cliente, en el caso de estas vistas la respuesta es un fichero HTML para mostrarse en el navegador:

- index. Respuesta index.html
- map. Respuesta map.html
- comparation. Respuesta comparation.html
- ranking. Respuesta ranking.html
- data. Respuesta data.html

Una vez definidas estas vistas se generan los ficheros HTML que contienen la estructura de cada una de las páginas definidas. Para ello se han usado unas plantillas de Bootstrap ya que dispone de estilos predefinidos para un diseño responsive. Los ficheros HTML se ubican en la carpeta de templates para tener una correcta organización del proyecto y evitar escribir rutas en las vistas.

Se establecen dos diseños en la aplicación denominados: index y dashboard. El primero será la página de bienvenida del proyecto en la que se muestra la información general del proyecto así como el contacto y que dará paso al dashboard. Esta página inicial (Figura 45) tiene como principal característica el estilo y el diseño atractivo para el usuario. Esta compuesta por secciones que pueden ser referenciadas en enlaces dentro de la misma página, de forma que en el encabezado de la página (que es fijo en toda la extensión) contiene los enlaces a las diferentes secciones para que haga un *scroll* a las mismas.

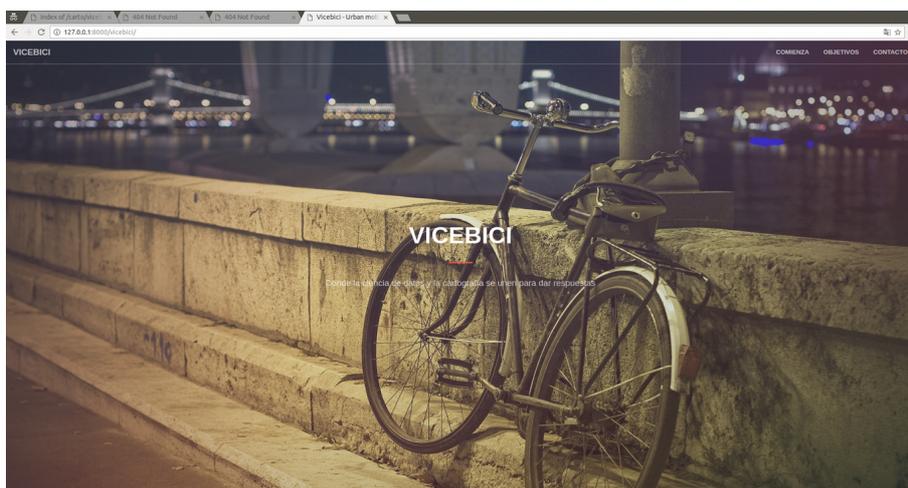


Figura 45: Página principal del proyecto. Se aprecia el encabezado las secciones en las que se compone la página inicial. Elaboración propia.

El diseño *dashboard* (Figura 46) corresponde a cuatro páginas: KPIs, comparación, ranking y mapa. Su contenido está compuesto por los diferentes datos y analíticas descritas anteriormente en la memoria. Este diseño se caracteriza por ser sobrio y funcional tiene como objetivo mostrar la información de manera clara. Para la navegación entre las páginas se ha ideado una barra lateral siempre visible.

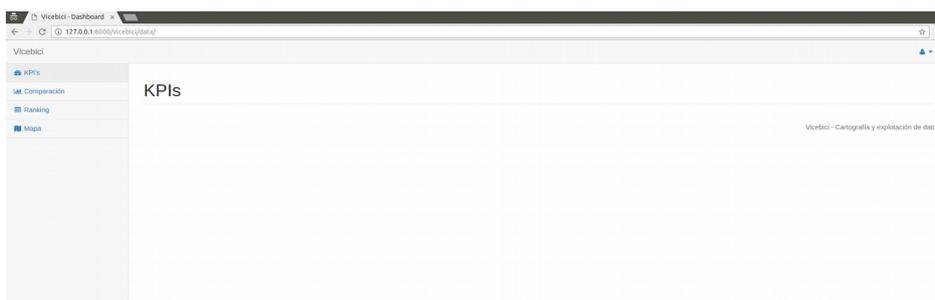


Figura 46: Diseño dashboard. Elaboración propia.

En este punto el desarrollo, los ficheros HTML pasan a llamarse plantillas de Django, esto es debido a que se va hacer uso de código que no es interpretable por un navegador, sino que es el framework quien lo traducirá generando dinámicamente parte de código HTML. Esto se realiza para definir URLs como los enlaces de la web o los ficheros estáticos.

Otra ventaja que tiene esta metodología de plantillas es la reutilización de código HTML mediante la función *extends*. Para llevar esto a cabo se han identificado las partes que van a ser comunes en las diferentes páginas del *dashboard* y se crea un fichero HTML en el cual se definen los bloques donde se van a introducir el código específico para cada página (Figura 47), de esta forma se desarrollan las funciones específicas para cada página pero se hereda siempre de una con el contenido que se repite. Esto es útil también para los ficheros estáticos, cada página hace uso de diferentes librerías específicas (leaflet, charts, css y JavaScript desarrollado), mientras que otras librerías serán comunes en todas las páginas (bootstrap y css).

```
.navbar-collapse">
  <span class="sr-only">Toggle navigation</span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="{% url 'vicebici:index' %}">Vicebici</a>
</div>
<!-- /.navbar-header -->
<ul class="nav navbar-top-links navbar-right">
  <!-- /.dropdown -->
  <li>
    {{ username }}
  </li>
  <li class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#">
      <i class="fa fa-user fa-fw"></i> <i class="fa fa-caret-down"></i>
    </a>
    <ul class="dropdown-menu dropdown-user">
      {% if user.is_authenticated %}
      <li><a href="{% url 'vicebici:logout' %}"><i class="fa fa-sign-out fa-fw"></i>
        Logout</a>
      </li>
      {% else %}
      <li><a href="{% url 'vicebici:login' %}"><i class="fa fa-sign-in fa-fw"></i>
        Login</a>
      </li>
      {% endif %}
    </ul>
  <!-- /.dropdown-user -->
</li>
<!-- /.dropdown -->
</ul>
<!-- /.navbar-top-links -->
{% if user.is_authenticated %}
<div class="navbar-default sidebar" role="navigation">
  <div class="sidebar-nav navbar-collapse">
    <ul class="nav" id="side-menu">
      <li>
        <a href="{% url 'vicebici:data' %}"><i class="fa fa-dashboard fa-fw"></i>
          RPT </a>
      </li>
    </ul>
  </div>
</div>
</div>
```

Figura 47: *Dashboard.html* con la definición de bloques “container” y “jss”. El resto de código HTML y de etiquetas script se heredan en los ficheros que hagan uso de la función *extend*. Elaboración propia.

Dentro de estas plantillas se ha creado unas secciones que son mostradas en el navegador en caso de que el cliente se acredite con un usuario y una contraseña. Como muestra la figura de arriba, el DOM contiene una condición que restringe una parte del HTML para que sea mostrada en caso de *loggeo*. El *login* se realiza en una página separada mediante una dirección de correo electrónico y contraseña asociada, para el que Django compruebe si existe en la base de datos. Hasta ahora la inscripción de usuarios se ha restringido, de forma que para dar de alta a un usuario se realiza por línea de comandos, una de las líneas de futuro es la gestión de inscripción de usuarios.

Por otro lado se han creado otras vistas que tendrán como objetivo realizar micro servicios web en forma de RESTful con formato GeoJSON. Estas vistas están diseñadas para enviar los datos referentes a los barrios de manera que las funciones habilitadas en el cliente sean mostradas de diferente forma (mapas, tablas o gráficas). El objetivo de estos servicios web es aislar las peticiones, de forma que el volumen de los datos no condicione el tiempo de carga de las páginas. Esto permite una buena experiencia de usuario dado que la carga del HTML es mucho más rápida. La metodología de carga de datos se explica en el Apartado 6.4.3.

Los servicios web desarrollados realizan una función de consulta en la base de datos, el resultado se serializa al formato GeoJSON. Estos servicios se encargan de que los datos que disponen de componente espacial estén en el sistema de referencia WGS84, expresados en coordenadas geográficas:

- `get_aparcabicis`: Este servicio devuelve la información de los aparcamientos de bicicleta públicos. El principal uso es la visualización en mapas y su geometría es de tipo punto.
- `get_barrios`: Este servicio devuelve los barrios junto con sus estadísticas. Este servicio se usa para mostrar los barrios, ranking y gráficas, la geometría es tipo polígono.
- `busca_barrios`: Este servicio permite realizar las búsquedas de barrios. Éste dispone de un parámetro, especificado mediante un formulario, para buscar aquellos barrios que se asemejen a las palabras escritas por el usuario.
- `get_estaciones`: Este servicio devuelve la información de las estaciones de Valenbisi. El principal uso es para su visualización en mapas y su geometría es de tipo punto.

### 6.3.3 Desarrollo cliente

En este apartado se va a explicar el desarrollo llevado a cabo para la capa de cliente o interfaz gráfica mediante un navegador web. Esta capa se basa en un desarrollo web mediante HTML5, CSS y JavaScript basado en una infraestructura de servidor definida en el apartado anterior. También se va a hacer referencia a las librerías JavaScript usadas para dotar de mayor funcionalidad a la interfaz web como Leaflet, JQuery, Charts.js y DataTables, y también a hojas de estilos CSS de Bootstrap.

El desarrollo en esta capa se ha centrado en la denominada parte con estilo *dashboard* (Apartado N Desarrollo servidor) en el cual se dota de funcionalidades específicas recogidas en el Apartado N Análisis funcional del sistema. En primer lugar, la navegación del usuario se ha diseñado mediante unos estilos intuitivos en las páginas y de uso común en la gran mayoría de las páginas, con los siguientes mecanismos:

- Encabezados con enlaces a secciones
- Barra lateral para navegación entre distintas páginas del dashboard
- Enlace a la página inicial situado en la parte superior izquierda
- Uso de iconos, estilos limpios e información sobre el contenido

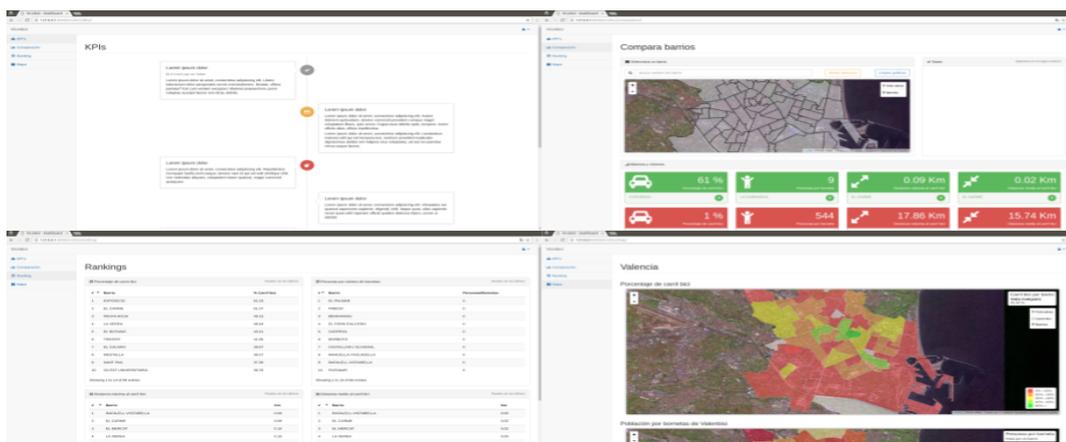


Figura 48: Composición de las páginas referentes al dashboard. Elaboración propia.

Cada página del dashboard está diseñada para mostrar la información, aunque se repita, de forma diferente. Para ello se ha ideado cuatro páginas.

- **KPIs o Índices**

El objetivo de esta página es dar información al usuario mostrando los índices calculados para el análisis junto con una descripción de cada uno de ellos. De esta forma cuando el usuario entra en la llamada parte *dashboard* es lo primero que se encuentra y contextualiza el análisis de movilidad con información de los distintos índices.



Figura 49: Página KPIs información sobre índices utilizados para el análisis de movilidad. Elaboración propia.

- **Comparación**

El objetivo de esta página es poder realizar comparativas de los datos de los barrios. En esta página se ha desarrollado una serie de funcionalidades que la hacen muy interactiva ya que combina el uso de la librería de mapas Leaflet y gráficas con Chart.js.

En esta página se ha hecho uso del servicio web *get\_barrios* para la obtención de las geometrías y datos de barrios, y *busca\_barrios* para realizar las búsquedas de los nombres de barrios. La carga de los datos se ha realizado asincrónicamente con Jquery con su módulo de Ajax y así obtener un GeoJSON. Se divide en tres bloques:

### 1. Mapa de selección y muestra de datos de un barrio.

Este bloque está compuesto por un mapa con los datos de los barrios y se ha dado un estilo homogéneo con transparencia, utilizando como mapa base la ortofoto del PNOA. También tiene un cuadro en el que se muestran los datos de los barrios.

Este mapa permite seleccionar las entidades de barrio, de forma que se resalta en amarillo la entidad y cuando se seleccione otra, la anterior pasa a tener el color de la capa inicial de barrios. También en el momento en que se hace esta selección, en el cuadro del lateral aparecen los datos referentes a población, kilómetros de carril bici, kilómetros de calle y número de bornetas que tiene el barrio (Figura 50).

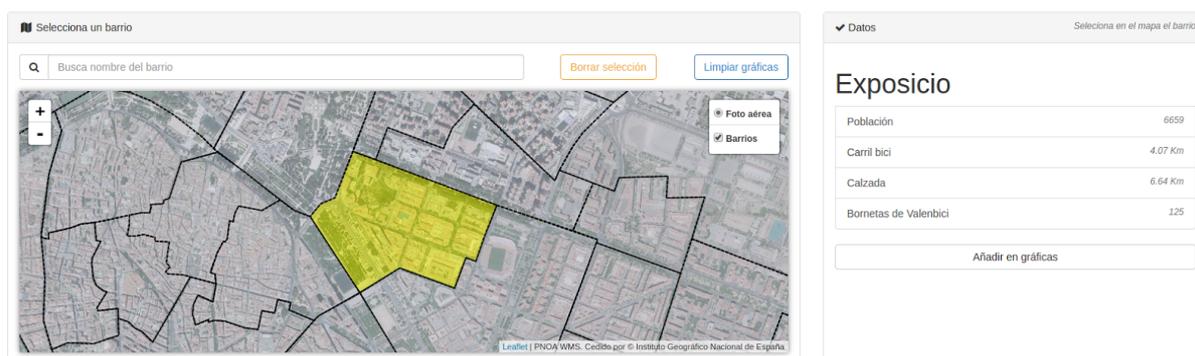


Figura 50: Mapa de selección de barrios y cuadro de datos del barrio seleccionado. Elaboración propia.

Se ha realizado un desarrollo en JavaScript para recoger los eventos que tienen que ver con la acción de selección en el mapa y la posterior inclusión de los datos en el cuadro lateral. Para ello se ha utilizado Jquery ya que permite la inserción de código HTML en el DOM, de forma que cada vez que se selecciona un barrio aparecen los datos del mismo junto con un botón con otras funcionalidades (este añade los datos a las gráficas del tercer bloque de esta página).

En el caso de las selecciones se ha hecho una clase JavaScript específica para controlar el barrio que se está seleccionando, de forma que las funciones que ejecuten alguna acción de selección deben de hacerse con dicha clase. Este controlador también se utiliza para las acciones con las gráficas del tercer bloque de esta página.

```

class HighCharts{
  constructor(){
    this.lsFeat = {};
    this.feat = 0;
  }
  addFeat(layer, feat, style){
    var response = false
    // check if exists the candidate
    if (this.lsFeat[feat] == undefined) {
      this.lsFeat[feat] = feat;
      this.highlight(layer, feat, style);
      response = true;
    }
    return response;
  };
  highlight(layer, idFeat, stylehigh){
    // highlight the layer of map
    layer.getLayer(idFeat).setStyle(stylehigh);
    this.feat = 0;
  };
  clearSelection(layer, stylehigh){
    // loop ids of selected feat of the map and set style and clear side panel
    jQuery.each(this.lsFeat, function(k, v){
      feature_map = lyr_barrios.getLayer(v);
      mymap._layers[v].setStyle(stylehigh);
    });
    $("#panel-derecho-body").html('')
    this.lsFeat = {};
  };
  clearHighlight(layer, stylehigh){
    if (this.feat > 0) {
      mymap._layers[this.feat].setStyle(stylehigh);
      this.feat = 0;
    }
  };
}
    
```

Figura 51: Clase HighCharts de control de los elementos de que están seleccionando en el mapa y gráficas. Elaboración propia.

Mediante la clase de la Figura 51 es posible controlar y sincronizar los elementos del mapa con las gráficas de tal forma que las acciones que hagan de disparador de modificación de los datos a visualizar tiene que pasar por dicha clase.

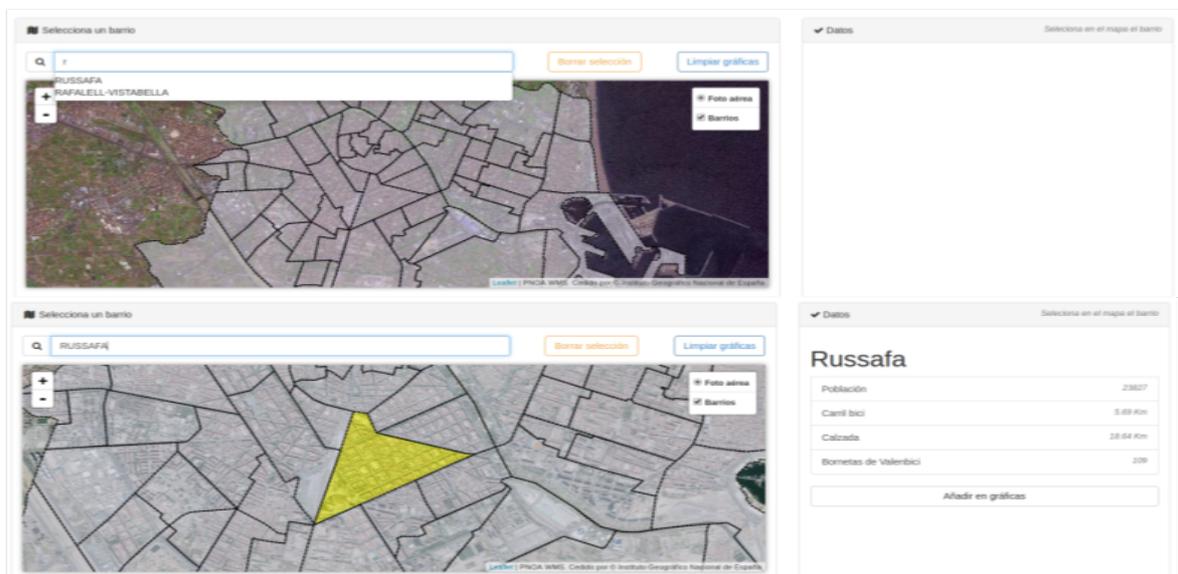


Figura 52: Búsqueda de un barrio (parte superior) selección del barrio Ruzafa (parte inferior). Elaboración propia.

Otra de las funcionalidades desarrolladas con JavaScript es un buscador de barrios dinámico, esto quiere decir que a medida que se escribe en el cuadro de texto, van apareciendo unos resultados que coinciden con el nombre del barrio en la base de datos. Al seleccionar el barrio deseado, el mapa realiza un zoom a la

entidad (encuadrando según su geometría) y lo dibuja como seleccionado, de forma que puede verse en el cuadro lateral su información (Figura 43).

Se han incluido en la parte superior dos botones con funcionalidades que limpian el mapa de las selecciones, estableciendo el estilo de polígonos inicial y dejando en blanco el cuadro lateral con los datos del barrio.

## 2. Índices con los máximos y mínimos

En este bloque se presentan los índices calculados de forma que se puede comparar los índices con sus máximos y mínimos. Esto ha supuesto un desarrollo en JavaScript ya que se busca para cada índice el máximo y mínimo, se inserta el dato en el DOM y se referencia el barrio al que pertenece (Figura 53). También cada cuadro de índice tiene un botón que realiza un zoom al mapa para saber cual es y muestra el cuadro lateral su información.



Figura 53: Índices con máximos y mínimos. Elaboración propia.

También se ha seleccionado un color para clasificar lo que representa un dato bueno (verde) o malo (rojo). Es decir, la fila superior no son los mínimos y la de abajo los máximos, puesto que el máximo de porcentaje de carril bici supone un buen índice para los ciclistas y es bueno para la movilidad, sin embargo una distancia alta entre un portal y el carril bici más cercano no es un buen índice para la movilidad en bici.

## 3. Gráficas de comparación



Figura 54: Mapa con los barrios seleccionados en azul que tienen los datos en las gráficas. Elaboración propia.

Para la comparación de los barrios mediante los datos de índices se recurre a las gráficas. Éstas tienen como característica principal que son actualizadas dinámicamente en función de las selecciones que se hagan en el

mapa. El funcionamiento se basa en que cuando un usuario selecciona un barrio (amarillo) y visualiza sus datos, en el cuadro lateral dispone de un botón que añade los datos de índices a las gráficas y en el mapa, este barrio pasa a tener un color azul Figura 45. La selección en azul permite al usuario ver los barrios que está comparando de forma que el análisis de las gráficas pueda tener una referencia espacial.

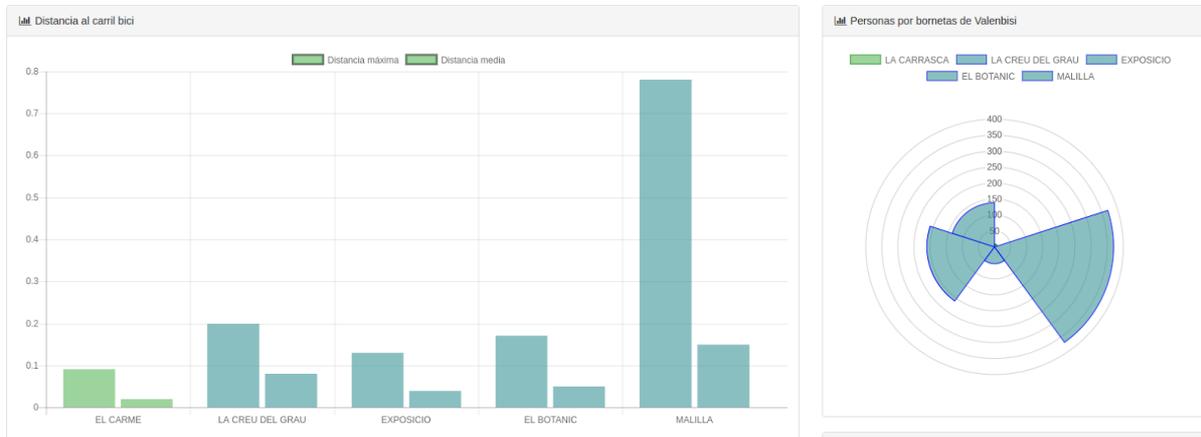


Figura 55: Gráfica de barras (izquierda) compara las distancias máximas y mínimas del carril bici más cercano en cada barrio. Gráfica polar (derecha) se representa la cantidad de personas por borneta de carril bici. Elaboración propia.

También se dispone de un botón para borrar tanto las gráficas como los barrios seleccionados. Este procedimiento se simplifica con la clase generada ya que se controlan los barrios seleccionados en amarillo de los que tienen la selección azul correspondiente a las gráficas.



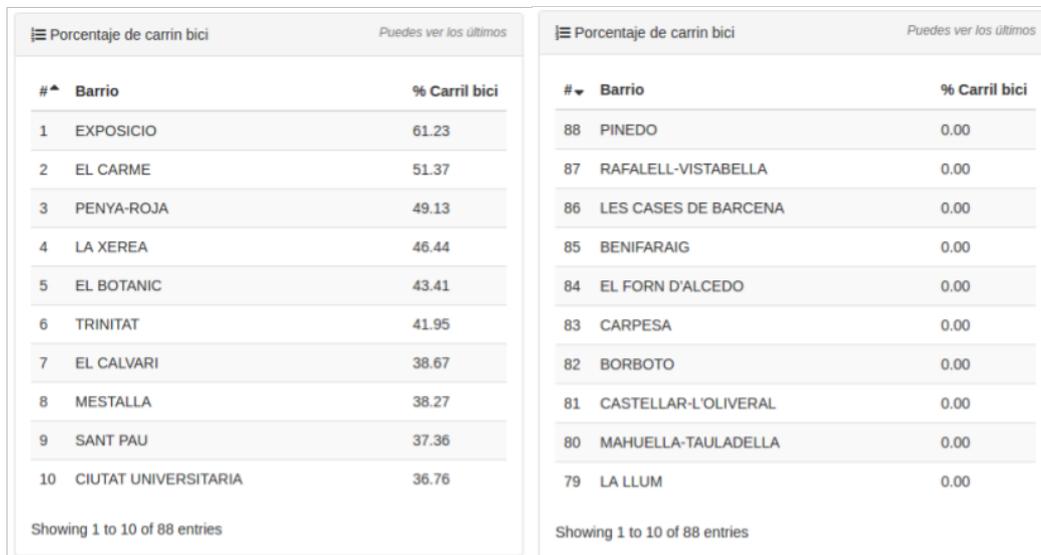
Figura 56: Gráfica de tarta comparando la longitud del carril bici. Muestra de desactivar elementos de una gráfica (izquierda). Elaboración propia.

Estas gráficas disponen también de funciones interactivas como *tooltips* o eliminar y añadir los datos que tenga cargados Figura 56. De esta forma se obtiene una buena experiencia para comparar distintos datos.

• **Ranking**

El objetivo de esta página es mostrar, en formato de tabla, los diez barrios con peor y mejor índice. Para ello se ha dispuesto de cuatro tablas realizadas con la librería DataTables, ya que dispone de funcionalidades de

filtrado, paginación y ordenar en función de columnas. Esta librería permite generar dinámicamente estas tablas de forma que son fácilmente actualizables (Figura 57).



#	Barrio	% Carril bici
1	EXPOSICIO	61.23
2	EL CARMÉ	51.37
3	PENYA-ROJA	49.13
4	LA XEREA	46.44
5	EL BOTANIC	43.41
6	TRINITAT	41.95
7	EL CALVARI	38.67
8	MESTALLA	38.27
9	SANT PAU	37.36
10	CIUTAT UNIVERSITARIA	36.76

#	Barrio	% Carril bici
88	PINEDO	0.00
87	RAFALELL-VISTABELLA	0.00
86	LES CASES DE BARCENA	0.00
85	BENIFARAIG	0.00
84	EL FORN D'ALCEDO	0.00
83	CARPESA	0.00
82	BORBOTO	0.00
81	CASTELLAR-L'OLIVERAL	0.00
80	MAHUELLA-TAULADELLA	0.00
79	LA LLUM	0.00

Figura 57: Tablas con el ranking del porcentaje de carril bici. Elaboración propia.

- **Mapas**

Esta página tiene como objetivo mostrar la información de los índices con mapas mediante Leaflet. Para ello se han creado dos mapas de coropletas con un graduado distinto de color (Figuras 58 y 59). Ambos disponen de funcionalidades parecidas, como mostrar el nombre del barrio y el dato cuando se pasa el cursor por encima de una entidad, y por otro lado se dispone de una leyenda. Otra funcionalidad es que al hacer clic en un barrio la vista se posiciona en la geometría.

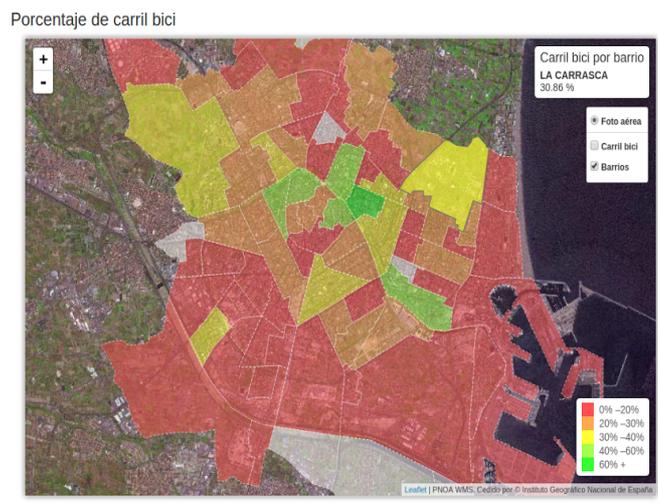


Figura 58: Mapa de coropletas del porcentaje de carril bici en Valencia. Elaboración propia.

Otra funcionalidad es la posibilidad de añadir datos relacionados con el índice que se quiere analizar. Esto se ha realizado habilitando un control de capas de forma que se puedan activar y desactivar. Estas capas disponen de la acción de hacer clic y ver su información asociada mediante un popup.

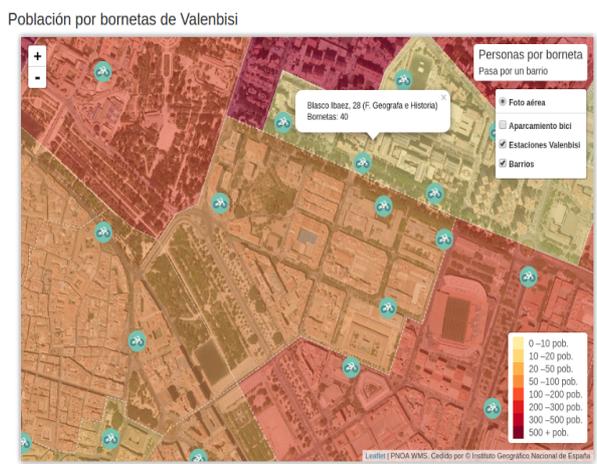


Figura 59: Mapa de coropletas del índice de población por bornetas de Valenbisi. Elaboración propia.

## 6.4 Despliegue de la aplicación

En este apartado se explica el proceso de despliegue que ha habido para acceder a la aplicación web a través de Internet. No es objeto del proyecto la explicación exhaustiva de un proceso de despliegue de una aplicación web, pero es de relevancia la posibilidad de que sea accesible esta aplicación para cualquier público. Con el objetivo de mostrar el estudio de movilidad urbana y poder realizar una prueba real del comportamiento de la aplicación en un entorno de producción se va a desplegar en un servidor.

Para tal fin en primer lugar se ha contratado un servidor dedicado en Kimsufi con las siguientes especificaciones:

- Procesador: Core™ i5-2300, 4 Cores / 4 Threads 2.8 GHz
- Almacenamiento: Disco duro 2 Tb
- Sistema operativo: Ubuntu server 16.04

Para el acceso al mismo y realizar todas las operaciones de configuraciones que van a ser descritas en este apartado se ha utilizado la conexión ssh mediante terminal. Esto es importante y supone tener una cuenta y contraseña para el usuario de acceso.

La primera configuración que se ha realizado es la instalación del firewall csf (ConfingServer Security Firewall) en el cual se deshabilita el acceso a los puertos más comunes con el fin de dotar de seguridad al servidor y que sólo sea accesible por ssh (habilitando el puerto 22). También se ha habilitado un puerto para el acceso a la base de datos mediante un túnel por el puerto 3333 en el que se necesita credenciales para el acceso.

Por otro lado ha sido necesaria la instalación de los componentes descritos en el desarrollo web como PostgreSQL con el módulo PostGIS. Se ha accedido de forma remota a la base de datos mediante el cliente pgAdmin (Figura 51) y de esta forma realizar la migración de datos. La migración consiste en generar una copia de la base de datos local para realizar una importación en la que hay alojada en el servidor. De esta forma la aplicación tendrá acceso a los datos sin tener que cambiar ninguna parte del código generado en el desarrollo en local, puesto que conserva el mismo modelo de datos.

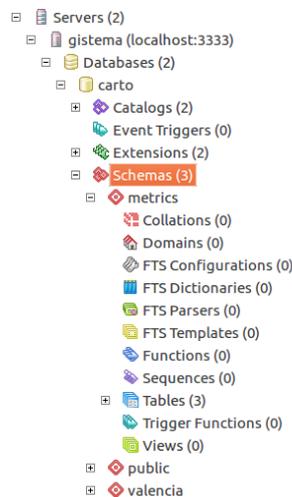


Figura 60: Acceso a la base de datos del servidor en remoto desde el cliente pgAdmin usando el puerto 3333. Previamente se ha debido generar el túnel de conexión. Elaboración propia.

Otro software del que se ha hecho uso en el servidor es git. Éste permite conectarse al repositorio de Bitbucket para descargar el proyecto web. Este software dispone de múltiples herramientas pero la primera descarga se realiza se ejecutando el comando *clone*.

```
git clone https://XXXXX@bitbucket.org/BArroquia/carto.git
```

Con el proyecto Django en el servidor se procede a realizar el despliegue de la aplicación. Para ello serán necesarias algunas comprobaciones previas y configuraciones con el servidor Apache. En primer lugar se va a realizar un test inicial al funcionamiento del proyecto y se va a hacer uso del *virtualenv* para instalar las librerías y versiones específicas para el intérprete Python utilizado en el desarrollo. Esto supone una ventaja ya que aísla al sistema principal de los módulos necesarios para esa aplicación (referentes a Python) y permite que no haya colisiones con otras librerías que usan otras aplicaciones en el servidor.

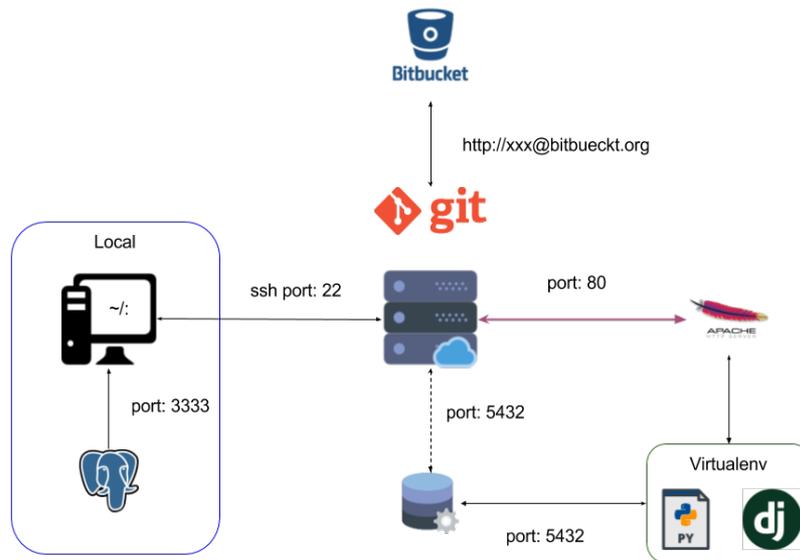


Figura 61: Diagrama de flujo de la configuración del servidor y las conexiones de los distintos módulos. Se ha utilizado un terminal para realizar las conexiones al servidor y crear un túnel ssh. Elaboración propia.

Una vez con el entorno virtual de Python en el servidor, se va a realizar una prueba de funcionamiento del proyecto descargado mediante la ejecución por línea de comandos del propio servidor que tiene Django. La comprobación se realiza haciendo la llamada al dominio del servidor mediante un navegador.

Una vez comprobado que funciona correctamente, se realizan las configuraciones para lanzar la aplicación web desde un servidor de producción como es Apache. Ubuntu tiene en su distribución dicho servidor por lo que solo es necesario realizar las configuraciones que permitan que el propio Apache pueda desplegar la aplicación. La configuración consiste en generar un fichero de configuración específico en la carpeta de `apache2` y definir el llamado *virtualhost*. En este fichero se establece el puerto por el que se conectará el cliente, la raíz del proyecto Django, ubicación de los ficheros estáticos y el módulo `wsgi.py` del proyecto que conectará con el instalado en Apache.

## 7 Conclusiones

El proyecto de análisis de movilidad en bicicleta en Valencia se ha completado de manera satisfactoria de forma que ayuda a comprender en profundidad el ámbito de la movilidad urbana desde la perspectiva de los usuarios de bicicletas. De este estudio se destaca como conclusiones:

- Con los datos disponibles en las plataformas estudiadas, se ha realizado un análisis y estudio de alternativas que sirven como base para generar una metodología de estudio de la movilidad en bicicleta. Se han definido índices que definen el estado de los elementos relacionados con este ámbito teniendo en cuenta su distribución en la ciudad de Valencia.
- Para la generación de este estudio se ha definido las fuentes de información disponibles y que son de libre disposición por los ciudadanos, como es el Portal de Datos Abiertos de Valencia. Se ha dado por manifiesto que es posible reutilizar datos abiertos para generar conocimiento y nuevas perspectivas que generen iniciativas sobre este ámbito. También es posible realizar este tipo de estudios en otros ámbitos, puesto que los portales de datos abiertos publican datos sobre medio ambiente, circulación o infraestructuras entre otros.
- El análisis de movilidad ha relacionado las variables relacionadas con la movilidad para obtener un mayor conocimiento de la ciudad de Valencia. En primer lugar se ha estudiado como es la distribución de los barrios en la ciudad de Valencia, y mediante el análisis calificar estos barrios y conocer los que son favorables para la práctica de la movilidad en bicicleta o cuales tienen carencias.
- Los resultados obtenidos tras el estudio y análisis de los datos ha generado una clasificación de los barrios con una calificación, la cual ha sido desarrollada y puede ser visible desde la aplicación web por cualquier usuario.
- La publicación de los datos y el análisis permite que los ciudadanos puedan conocer los datos de una forma sencilla y transparente. La aplicación web permite superar las barreras que existen entre los portales de datos abiertos (que sirven los datos en formatos que necesitan un tratamiento técnico) y los ciudadanos. También permite realizar análisis con las diferentes funcionalidades desarrolladas (visualización con gráficas y mapas, comparaciones y consultas a barrios). La aplicación permite realizar análisis al usuario no dirigidos mediante los índices y los datos que generan los índices para que se puedan tener conclusiones personales.

### 7.1 Líneas de investigación

Aunque los objetivos del proyecto fin de carrera se han cumplido satisfactoriamente, existe todavía un margen de mejora añadiendo algunas funcionalidades. Algunas de las líneas de mejora son:

- Generación de procesos automatizados para la descarga de la información e importación en base de datos. Crear una metodología en la que se analicen las posibilidades en cuanto a tecnologías web como APIs en las que la aplicación pueda conectarse, de forma que se pueda realizar también actualizaciones periódicas de los datos. Esto dotaría de más dinamismo al sistema para obtener métricas actualizadas e incluso generar históricos.
- Creación de nuevos índices sobre movilidad en bicicleta con el uso de topología de red con pgrouting. Mediante esta técnica se puede obtener un cálculo de distancias y rutas en función de unas posiciones determinadas. De esto pueden resultar análisis como rutas saludables y ecológicas o cuanto impacto puede tener en un barrio un carril bici en cuestión de tráfico y su importancia en la red.
- Desarrollo de un apartado de aportaciones y comentarios para conocer el impacto que puede tener a los ciudadanos esta propuesta y medir la importancia que tiene.

## 8 Bibliografía

- [1] Pujadas Rúbies, I., Bayona Carrasco, J., García Coll, A., Gil Alonso, F., López Villanueva, C., Sánchez Aguilera, D. y Vidal Bendito, T. “Población y espacios urbanos”. Departament de Geografia Humana de la UB y Grupo de población de la AGE. <[http://www.ub.edu/congreso\\_poblacion/docs/actas.pdf](http://www.ub.edu/congreso_poblacion/docs/actas.pdf)> [Consulta 1 Junio 2017]
- [2] Instituto Nacional de Estadística (INE) (2016). Indicadores Urbanos (Urban Audit). <<http://www.ine.es/prensa/np960.pdf>> [Consulta 1 Junio 2017]
- [3] Madeu Àrias, J. (2017). “Sevilla y Valencia abren sus calles a las bicis mientras que Madrid permanece a la cola” en 20minutos. <<http://www.20minutos.es/noticia/3023963/0/sevilla-valencia-ciclismo-urbano-bicis-madrid/#xtor=AD-15&xts=467263>> [Consulta 20 mayo 2017]
- [4] Rodríguez Bustamante, P. (2017). “Open Data y movilidad, no sólo transporte” en Momento digital. <<http://momento.digital/open-data-movilidad-no-solo-transporte/>> [Consulta 1 Junio 2017]
- [5] Gómez, D. (2015). “Así fue como Ámsterdam se hizo la ciudad de las bicicletas“ en El Tiempo. <<http://www.eltiempo.com/archivo/documento/CMS-16263915>> [Consulta 1 Junio 2017]
- [6] Organización de Consumidores y Usuarios (OCU) (2017). “OCU señala a Sevilla y Valencia como las mejores ciudades para los ciclistas, Madrid la peor” en OCU. <<https://www.ocu.org/organizacion/prensa/notas-de-prensa/2017/encuesta-bicis-290317>> [Consulta 2 Junio 2017]
- [7] Organización de Consumidores y Usuarios (OCU) (2013). “Circular en bici” en OCUCompra Maestra vol 384 p26 - 30. <<https://www.ocu.org/toda-la-informacion?type=magazine-articles&magazine=ocu+compra+maestra&year=2013>> [Consulta 2 Junio 2017]
- [8] Ayuntamiento de Valencia. *Portal Transparència i Dades Obertes*. <<http://gobiernoabierto.valencia.es/va/>> [Consulta 20 Agosto 2017]
- [9] Ayuntamiento de Sevilla. *Portal de Datos Abiertos del Ayuntamiento de Sevilla*. <<http://datosabiertos.sevilla.org/>> [Consulta 2 Junio 2017]
- [10] Ayuntamiento de Barcelona. *Open Data BCN | Servicio de datos abiertos del Ajuntament de Barcelona*. <<http://opendata-ajuntament.barcelona.cat/es>> [Consulta 2 Junio 2017]
- [11] Ayuntamiento de Madrid. *En portada - Portal de datos abiertos del Ayuntamiento de Madrid*. <<http://datos.madrid.es/portal/site/egob>> [Consulta 2 Junio 2017]
- [12] Ayuntamiento de Adelaide. *Invest Adelaide* <<http://investadelaide.com.au/>> [Consulta 2 Junio 2017]
- [13] Red.es. *datos.gob.es*. <<http://datos.gob.es/>> [Consulta 2 Junio 2017]

- [14] Ayuntamiento de Valencia. *Portal del Ayuntamiento de la ciudad de Valencia*. <[https://www.valencia.es/ayuntamiento/home.nsf/\(Portadas1\)/\\$first?opendocument&lang=1](https://www.valencia.es/ayuntamiento/home.nsf/(Portadas1)/$first?opendocument&lang=1)> [Consulta 2 Junio 2017]
- [15] Bartual Roig, J (2017). “Valenbisi pierde el 50% de los usuarios en tres años con un descenso del 23% en los viajes” en Levante EMV. <<http://www.levante-emv.com/valencia/2017/01/04/valenbisi-pierde-50-usuarios-tres/1511374.html>> [Consulta 2 Junio 2017]
- [16] Instituto Geográfico Nacional (IGN). *Instituto Geográfico Nacional*. <<http://www.ign.es/web/ign/portal/inicio>> [Consulta 2 Junio 2017]
- [17] Instituto Cartográfico Valenciano. *ICV - Generalitat Valenciana*. <<http://www.icv.gva.es/ca/inicio>> [Consulta 3 Junio 2017]
- [18] Instituto Nacional de Estadística. *Instituto Nacional de Estadística. (Spanish Statistical Office)*. <<http://www.ine.es/welcome.shtml>> [Consulta 3 Junio 2017]
- [19] Bicimetro. *Inicio*. <<http://bicimetro.plumtic.es/>> [Consulta 1 Junio 2017]
- [20] AENOR (2017). *Accesibilidad universal en las Ciudades Inteligentes*. UNE 178105:2017. Madrid: AENOR. [Consulta 20 Agosto 2017]
- [21] Martínez Llario, J. C. (2014). *PostGIS 2 Análisis Espacial Avanzado*. Charleston, SC, USA.
- [22] Idris, I (2014). *Python data analysis*. Packt publishing, UK.
- [23] Ramaswami, A., Russell, A.G., Culligan, P.J., Rahul Sharma, K., Kumar, E. (2016). *Meta-principles for developing smart, sustainable, and healthy cities*. Science vol. 352. Issue 6288. p. 940-943.

## 9 Anexos

### 9.1 Tabla de resultados

Nombre barrio	Índices				Normalización				Calificación
	Porcentaje carril bici	Población por borneta	Máxima distancia (metros)	Distancia media (metros)	Porcentaje carril bici normalizado	Población por borneta normalizado	Máxima distancia normalizado	Distancia media normalizado	
EXPOSICIO	61,23	53,27	127,89	38,42	10,00	9,02	8,72	9,04	9,20
EL CARME	51,37	64,05	90,44	21,91	8,39	8,82	9,10	9,45	8,94
LA XEREA	46,44	63,58	103,69	26,27	7,58	8,83	8,96	9,34	8,68
PENYA-ROJA	49,13	93,04	190,80	56,04	8,02	8,29	8,09	8,60	8,25
TRINITAT	41,95	106,01	156,81	62,71	6,85	8,05	8,43	8,43	7,94
LA SEU CIUTAT	28,46	58,83	117,91	33,85	4,65	8,92	8,82	9,15	7,89
UNIVERSITARIA	36,76	19,08	243,38	74,56	6,00	9,65	7,57	8,14	7,84
EL MERCAT	23,02	98,63	96,49	24,46	3,76	8,19	9,04	9,39	7,59
EL BOTANIC	43,41	210,35	168,70	51,95	7,09	6,13	8,31	8,70	7,56
EL CALVARI	38,67	237,80	121,34	44,00	6,32	5,63	8,79	8,90	7,41
MESTALLA	38,27	115,74	246,75	95,52	6,25	7,87	7,53	7,61	7,32
LES TENDETES EL PLA DEL	34,49	259,85	116,19	43,30	5,63	5,22	8,84	8,92	7,15
REMEI	26,47	54,07	240,66	106,03	4,32	9,01	7,59	7,35	7,07
LA GRAN VIA	28,19	98,39	235,24	88,57	4,60	8,19	7,65	7,79	7,06
ARRANCAPINS	26,48	117,27	272,68	69,28	4,33	7,84	7,27	8,27	6,93
MARXALENES	20,33	132,52	183,00	66,84	3,32	7,56	8,17	8,33	6,85
CAMPANAR	22,06	68,14	276,83	97,16	3,60	8,75	7,23	7,57	6,79
LA VEGA BAIXA	20,04	145,31	195,02	76,50	3,27	7,33	8,05	8,09	6,68
RUSSAFA	30,53	218,60	322,80	58,94	4,99	5,98	6,77	8,53	6,57
SANT FRANCESC	8,32	36,17	222,95	88,80	1,36	9,33	7,77	7,78	6,56

CABANYAL- CANYAMELAR	17,25	98,23	285,96	76,35	2,82	8,19	7,14	8,09	6,56
SANT PAU NA ROVELLA	37,36	75,80	546,12	120,60	6,10	8,61	4,54	6,99	6,56
SANT ISIDRE CIUTAT DE LES ARTS I DE LES CIENCIAS	20,07	166,10	222,32	76,84	3,28	6,95	7,78	8,08	6,52
BENICALAP	30,86	153,22	370,87	110,92	5,04	7,18	6,29	7,23	6,44
NOU MOLES	25,92	40,00	357,36	187,77	4,23	9,26	6,43	5,31	6,31
TORMOS	24,62	196,16	333,80	90,97	4,02	6,39	6,66	7,73	6,20
AIORA	21,16	205,46	264,86	92,82	3,46	6,22	7,35	7,68	6,18
LA PETXINA	16,68	211,20	217,76	79,91	2,72	6,12	7,82	8,00	6,17
LA ROQUETA	23,91	173,27	387,12	98,57	3,90	6,81	6,13	7,54	6,10
MONT-OLIVET	23,54	295,12	238,72	70,75	3,84	4,57	7,61	8,23	6,07
CIUTAT JARDI	12,14	73,42	288,93	140,91	1,98	8,65	7,11	6,48	6,06
LA RAIOSA	16,67	197,81	248,23	101,11	2,72	6,36	7,52	7,47	6,02
EN CORTS	22,42	150,30	396,54	117,75	3,66	7,24	6,03	7,06	6,00
TRES FORQUES	18,19	228,76	268,99	93,08	2,97	5,79	7,31	7,67	5,94
BENIFERRI	22,72	317,95	237,23	83,27	3,71	4,15	7,63	7,92	5,85
FAVARA	15,32	191,40	273,55	118,43	2,50	6,48	7,26	7,04	5,82
EL PILAR	27,11	29,50	610,32	186,52	4,43	9,46	3,90	5,34	5,78
MORVEDRE	11,95	101,14	345,35	150,39	1,95	8,14	6,55	6,24	5,72
L'ILLA PERDUDA	26,67	0,00	122,78	27,26	4,36	0,00	8,77	9,32	5,61
TORREFIEL	17,40	340,17	215,27	91,15	2,84	3,75	7,85	7,72	5,54
LA FONTSANTA	9,94	214,10	302,54	129,48	1,62	6,06	6,97	6,76	5,36
SANT LLORENS	16,98	270,51	317,44	132,39	2,77	5,03	6,83	6,69	5,33
LA CREU DEL GRAU	5,99	52,07	379,51	200,40	0,98	9,04	6,20	4,99	5,30
BENIMACLET	26,39	139,39	744,22	124,89	4,31	7,44	2,56	6,88	5,30
CAMI DE VERA	10,72	369,63	195,50	79,81	1,75	3,20	8,05	8,00	5,25
	15,87	263,15	383,29	117,89	2,59	5,16	6,17	7,05	5,24
	24,64	175,97	509,92	192,95	4,02	6,76	4,90	5,18	5,22

JAUME ROIG	7,34	320,00	231,33	95,12	1,20	4,12	7,69	7,62	5,16
BETERO	24,85	0,00	221,78	56,50	4,06	0,00	7,78	8,59	5,11
CAMI FONDO	0,00	271,47	190,63	107,02	0,00	5,01	8,09	7,32	5,11
LA CARRASCA	30,86	9,20	1002,00	182,59	5,04	9,83	0,00	5,44	5,08
SAFRANAR	7,40	177,46	362,35	167,71	1,21	6,74	6,38	5,81	5,03
L'AMISTAT	7,06	157,59	417,97	193,95	1,15	7,10	5,82	5,15	4,81
LA CREU									
COBERTA	14,29	308,75	342,99	180,62	2,33	4,32	6,57	5,48	4,68
L'HORT DE									
SENBRE	15,01	461,65	271,65	102,76	2,45	1,51	7,28	7,43	4,67
ALBORS	10,26	107,73	469,58	254,88	1,68	8,02	5,30	3,63	4,66
MALILLA	11,12	137,96	777,40	149,08	1,82	7,46	2,23	6,27	4,44
LA FONTETA									
S.LLUIS	5,31	94,37	534,40	263,40	0,87	8,27	4,66	3,42	4,30
ELS ORRIOLS	14,16	265,28	748,89	117,29	2,31	5,12	2,51	7,07	4,25
LA MALVA-ROSA	8,43	128,09	565,65	256,02	1,38	7,65	4,34	3,60	4,24
EL GRAU	14,69	176,04	1996,51	89,57	2,40	6,76	0,00	7,76	4,23
CIUTAT FALLERA	6,53	182,35	457,81	248,60	1,07	6,65	5,42	3,78	4,23
SANT MARCEL.LI	6,74	261,08	449,19	199,36	1,10	5,20	5,51	5,02	4,21
PATRAIX	15,99	543,91	334,80	100,89	2,61	0,00	6,65	7,48	4,19
VARA DE QUART	5,12	146,53	735,79	238,03	0,84	7,31	2,64	4,05	3,71
CAMI REAL	9,99	103,66	720,46	359,51	1,63	8,09	2,80	1,01	3,38
SANT ANTONI	0,00	484,65	325,69	192,98	0,00	1,09	6,74	5,18	3,25
NATZARET	4,37	205,90	639,04	301,28	0,71	6,21	3,61	2,47	3,25
LA PUNTA	12,27	87,50	1401,23	365,66	2,00	8,39	0,00	0,86	2,81
SOTERNES	0,00	168,28	822,05	349,11	0,00	6,91	1,78	1,27	2,49
BENIMAMET	2,30	441,14	790,03	324,74	0,38	1,89	2,10	1,88	1,56
FAITANAR	18,73	0,00	982,74	295,09	3,06	0,00	0,17	2,62	1,46
LA LLUM	0,00	253,40	986,00	604,42	0,00	5,34	0,14	0,00	1,37
LA TORRE	1,72	0,00	775,65	484,38	0,28	0,00	2,24	0,00	0,63
POBLE NOU	4,26	0,00	1275,88	613,58	0,70	0,00	0,00	0,00	0,17

RAFALELL- VISTABELLA	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
BORBOTO	0,00	0,00	2466,74	1407,85	0,00	0,00	0,00	0,00	0,00	0,00
PINEDO	0,00	0,00	5167,93	1484,07	0,00	0,00	0,00	0,00	0,00	0,00
EL PALMAR	0,00	0,00	14050,10	13624,10	0,00	0,00	0,00	0,00	0,00	0,00
CASTELLAR- L'OLIVERAL	0,00	0,00	2415,49	1860,46	0,00	0,00	0,00	0,00	0,00	0,00
EL SALER	0,00	0,00	12775,75	6384,24	0,00	0,00	0,00	0,00	0,00	0,00
EL FORN										
D'ALCEDO	0,00	0,00	1765,19	1271,06	0,00	0,00	0,00	0,00	0,00	0,00
CARPESA	0,00	0,00	2808,02	1829,88	0,00	0,00	0,00	0,00	0,00	0,00
MASSARROJOS	0,00	0,00	6230,04	4059,52	0,00	0,00	0,00	0,00	0,00	0,00
EL PERELLONET	0,00	0,00	17856,53	15744,74	0,00	0,00	0,00	0,00	0,00	0,00
BENIFARAIG	0,00	0,00	3218,02	2865,32	0,00	0,00	0,00	0,00	0,00	0,00
LES CASES DE BARCENA	0,00	0,00	2685,79	1843,39	0,00	0,00	0,00	0,00	0,00	0,00
MAHUELLA- TAULADELLA	0,00	0,00	4668,28	4060,93	0,00	0,00	0,00	0,00	0,00	0,00