



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Herramienta web para visualización de datos de uso de bicicletas en ciudades

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

*Autor:* Vicent Pérez Castelló

*Tutor:* Cèsar Ferri Ramírez

Curso 2018-2019



# Resum

Els sistemes de compartició de bicicletes ofereixen als ciutadans la possibilitat de desplaçar-se per una xarxa d'estacions, poden elegir lliurement els punts de partida i de destí. Aquest tipus de sistemes estan guanyant popularitat, ja que permeten combatre l'ús intensiu de vehicles, beneficiant tant la salut dels ciutadans com el medi ambient.

Aquest projecte aborda la implementació d'una ferramenta web interactiva per a la visualització de l'ús de les bicicletes en ciutats. La ferramenta permet l'exploració d'un corpus de dades que ha sigut recopilat en 3584 estacions en 27 ciutats d'11 distints països, durant un període de 7 mesos, gràcies a un servei API públic proporcionat per JCDecaux, una de les empreses més influents en el sector de compartició de bicicletes. Aquestes dades es complementen amb un altre corpus de dades de l'estat meteorològic en el mateix transcurs de temps, enriquint d'aquesta manera l'anàlisi de la demanda.

Les visualitzacions ens permeten encontrar patrons d'ús, tant diaris com setmanals, i relacions entre la demanda i el factor meteorològic en cada una de les ciutats. Així mateix, ens permet comparar perfils diaris i setmanals de diferents ciutats i/o estacions. A més a més, ens ofereix la possibilitat de visualitzar l'estat de les estacions en temps real, poden consultar per part de l'usuari el nombre de bicicletes i estands disponibles en cadascuna de les estacions.

Per a finalitzar es realitza un exemple d'ús de la ferramenta on s'exploren de manera visual les dades de les estacions de València.

Implicacions pràctiques, limitacions i futures direccions d'ampliació de la ferramenta conclouen el projecte.

**Paraules clau:** Anàlisi de dades, R, Shiny, ggplot, Sistemes de compartició de bicicletes, Visualització

---

# Resumen

Los sistemas de compartición de bicicletas ofrecen a los ciudadanos la posibilidad de desplazarse por una red de estaciones, pudiendo elegir libremente los puntos de partida y destino. Este tipo de sistemas están ganando popularidad ya que permite combatir el uso intensivo de vehículos, beneficiando tanto la salud de los ciudadanos como el medio ambiente.

Este proyecto aborda la implementación de una herramienta web interactiva para la visualización del uso de las bicicletas en ciudades. La herramienta permite la exploración de un corpus de datos que ha sido recopilado en 3584 estaciones en 27 ciudades de 11 distintos países, durante un periodo de 7 meses, gracias a un servicio API público proporcionado por JCDecaux, una de las empresas más influyentes en el sector de compartición de bicicletas. Estos datos se complementan con otro corpus de datos del estado meteorológico en el mismo transcurso de tiempo, enriqueciendo así el análisis de la demanda.

Las visualizaciones nos permiten encontrar patrones de uso, tanto diarios como semanales, y relaciones entre la demanda y el factor meteorológico en cada una de las ciudades. A su vez, nos permite la comparación entre los perfiles diarios y semanales de diferentes ciudades y/o estaciones. Además, nos ofrece la posibilidad de visualizar el estado de las estaciones en tiempo real pudiendo consultar por parte del usuario el número de bicicletas y stands disponibles en cada una de las estaciones.

Para finalizar se realiza un ejemplo de uso de la herramienta donde se exploran de manera visual los datos de las estaciones de Valencia.

Implicaciones prácticas, limitaciones y futuras direcciones de ampliación de la herramienta concluyen el proyecto.

**Palabras clave:** Análisis de datos, R, Shiny, ggplot, Sistemas de compartición de bicicletas, Visualizaciones

---



# Abstract

The bicycle-sharing systems offer citizens the possibility of travelling throughout a network of stations, allowing them to choose freely both departure and destination points. This kind of systems is gaining popularity as it helps to face the intensified use of vehicles, benefiting both the health of citizens and the environment.

This project addresses the implementation of an interactive web tool for visualizing the use of bicycles in cities. The tool gives us the chance to explore a corpus of data that has been collected at 3584 stations in 27 cities in 11 different countries, during a period of 7 months. This was possible due to the public API service provided by JCDecaux, one of the most influential companies in the bicycle-sharing sector. The data are complemented with another corpus of data containing the meteorological state during the same period of time, thus enriching the analysis of the demand.

The visualizations enable us to find usage patterns, both daily and weekly, as well as relationships between demand and the meteorological factor in each city. Moreover, using the tool we can compare the daily and weekly profiles of different cities and/or stations. In addition, it offers us the possibility of viewing the status of each station in real time. Consequently, the user is able to check the availability of stands and bicycles on each of the stations.

To finish, a usage example of the tool is made, where the data of the stations of Valencia are explored in a visual way. Practical implications, limitations and future directions of the tool conclude the project.

**Key words:** Data Analysis, R, Shiny, ggplot, Bicycle-Sharing systems, Visualization

---



# Índice general

---

<b>Índice general</b>	VII
<b>Índice de figuras</b>	IX
<b>Índice de tablas</b>	X
<hr/>	
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	2
1.2 Objetivos . . . . .	3
1.3 Estructura de la memoria . . . . .	4
<b>2 Estado del arte</b>	<b>5</b>
2.1 Soluciones actuales . . . . .	5
2.2 Propuesta . . . . .	7
<b>3 Análisis del problema</b>	<b>9</b>
3.1 Análisis de requisitos . . . . .	9
3.2 Análisis de tecnologías . . . . .	14
3.3 Datos disponibles . . . . .	15
<b>4 Diseño de la aplicación</b>	<b>17</b>
4.1 Arquitectura del software . . . . .	17
4.2 Almacenamiento de los datos . . . . .	18
4.3 Diseño de la interfaz . . . . .	19
4.4 Tecnología utilizada . . . . .	25
<b>5 Implementación</b>	<b>29</b>
5.1 Desarrollo de la interfaz de usuario . . . . .	29
5.2 Funciones para el procesado de datos . . . . .	30
5.3 Sección información ciudades . . . . .	31
5.4 Sección información estaciones . . . . .	34
5.5 Sección comparación perfiles . . . . .	36
5.6 Sección tiempo real . . . . .	38
<b>6 Ejemplo de análisis de datos</b>	<b>39</b>
<b>7 Conclusiones y trabajo futuro</b>	<b>49</b>
<b>Bibliografía</b>	<b>51</b>
<hr/>	
Apéndice	
<b>A Obtención del código fuente de la herramienta</b>	<b>53</b>



# Índice de figuras

---

4.1	Arquitectura del proyecto	17
4.2	Organización directorios y ficheros en la herramienta	18
4.3	Diseño base de la herramienta	19
4.4	Apartado información general ciudades	20
4.5	Apartado información meteorológica	20
4.6	Apartado información demanda junto con meteorológica	21
4.7	Apartado visualización perfiles ciudades	21
4.8	Apartado informativo de las estaciones	22
4.9	Apartado visualización demanda según fecha	22
4.10	Apartado visualización perfiles estación	23
4.11	Comparación perfiles ciudades	23
4.12	Comparación perfiles estaciones	24
4.13	Primera caja sección ciudades	24
4.14	Logo librería Shiny	25
4.15	Tipos de objetos en la programación reactiva de Shiny	26
4.16	Relación simple de reactividad	26
4.17	Ejemplo reactividad en Shiny	27
5.1	Código unificación ficheros secciones	29
5.2	Diagrama de relaciones entre los elementos del apartado información general de la ciudad	32
5.3	Diagrama de relaciones del apartado meteorología	33
5.4	Diagrama relaciones apartado información del uso de bicicletas	34
5.5	Relaciones apartado información estaciones	35
5.6	Diagrama relaciones apartado información demanda estaciones	36
5.7	Diagrama relaciones apartado comparación perfiles de ciudades	37
5.8	Diagrama relaciones apartado comparacion perfiles estaciones	37
5.9	Relaciones entre los elementos del apartado tiempo real	38
6.1	Información general de la ciudad de Valencia	39
6.2	Gráfico de temperatura de la ciudad de Valencia	40
6.3	Gráfico de lluvias de la ciudad de Valencia	40
6.4	Rosa de vientos de la ciudad de Valencia	41
6.5	Correlación entre la temperatura y la demanda en Valencia	41
6.6	Información diaria de la demanda junto al factor meteorológico en Valencia	42
6.7	Perfil diario de la demanda en la ciudad de Valencia	42
6.8	Perfil semanal de la demanda en la ciudad de Valencia	43
6.9	Información de las estaciones de Valencia	43
6.10	Estación dentro de la Universitat Politècnica de València	44
6.11	Visualización de la demanda en el día 15 de Enero del 2015	44

6.12	Visualización de la demanda de la semana del 12 de Enero del 2015	45
6.13	Perfil diario de la estación de la Universitat Politècnica de València	45
6.14	Perfil semanal de la estación de la Universitat Politècnica de València	46
6.15	Comparación del perfil diario entre Valencia y Toulouse . . . . .	46
6.16	Comparación del perfil diario entre Valencia y Sevilla . . . . .	47
6.17	Comparación de perfiles entre dos estaciones cercanas de Valencia .	47
6.18	Comparación de perfiles entre dos estaciones distantes de Valencia	48
6.19	Visualización en tiempo real del estado de la estación de la Univer- sitat Politècnica de València . . . . .	48

## Índice de tablas

---

3.1	REQF1: Diseño base de la herramienta . . . . .	9
3.2	REQF2: Menú lateral izquierdo . . . . .	10
3.3	REQF3: Información de la ciudad seleccionada . . . . .	10
3.4	REQF4: Ranking uso ciudad . . . . .	10
3.5	REQF5: Gráficos meteorológicos ciudad . . . . .	10
3.6	REQF6: Visualización uso de bicicleta y tiempo . . . . .	11
3.7	REQF7: Perfiles diarios y semanales ciudad . . . . .	11
3.8	REQF8: Comparar perfiles estaciones y ciudades . . . . .	11
3.9	REQF9: Mapa de estaciones según demanda . . . . .	11
3.10	REQF10: Información detallada de la estación . . . . .	12
3.11	REQF11: Gráfico demanda estación . . . . .	12
3.12	REQF12: Perfil semanal y diario estación . . . . .	12
3.13	REQF13: Mapa de estaciones en tiempo real . . . . .	12
3.14	REQF14: Información estación en tiempo real . . . . .	13
3.15	REQNF1: Número de usuarios . . . . .	13
3.16	REQNF2: Obtención datos API . . . . .	13
3.17	REQNF3: Carga de visualizaciones . . . . .	13
3.18	REQNF4: Tiempo máximo de aprendizaje . . . . .	14
3.19	REQNF5: Interfaces gráficas . . . . .	14
3.20	REQNF6: Navegadores disponibles . . . . .	14

---

---

# CAPÍTULO 1

## Introducción

---

Una ciudad inteligente es una área urbana evolucionada donde se crea un desarrollo económico sostenible y de alta calidad dentro de múltiples áreas clave, como son la economía, la movilidad, el medio ambiente, la personas y el gobierno. Las ciudades se sensorizan con el objetivo de aumentar la conectividad, automatización y coordinación entre proveedores y consumidores [1].

La simbiosis entre sostenibilidad y movilidad provoca adentrarse en un interesante campo como es el sistema de compartición de bicicletas. Sistema que está ganando gran popularidad como alternativa al intensivo uso de vehículos contaminantes [2].

Si nos remontamos a los inicios, el primer sistema de compartición de bicicletas apareció en 1965 concretamente, en la ciudad holandesa de Amsterdam. Consistía en una serie de bicicletas de color blanco que estaban distribuidas por la ciudad, las cuales se podían utilizar sin ningún tipo de coste asociado [7]. Este primer sistema, no tuvo éxito debido al vandalismo puesto que había un deterioro o robo de bicicletas continuado. A partir de entonces, se han ido perfeccionando día a día, utilizando las nuevas tecnologías para prevenir robos y tener sistemas más seguros y eficientes [2].

Existen diferentes modelos como por ejemplo, de largo plazo, de uso por moneda o de libre uso. En este proyecto nos centramos en los sistemas de corto plazo que permiten al usuario coger prestada una bicicleta por un periodo de tiempo determinado, sin importar la estación origen y destino. Las bicicletas están sujetas mediante un seguro que se libera cuando el cliente se identifica en el sistema. A partir de este momento se dispone de 30 minutos de uso gratuito, aplicándose una tarifa monetaria si se sobrepasa [11].

El punto innovador que queremos introducir va a mejorar uno de los principales retos a los que se enfrentan las compañías que sustentan estos sistemas, el balanceo de bicicletas. Pueden darse casos donde una estación queda completamente llena o vacía. Para ello, existen camiones que van moviendo las bicicletas desde estaciones prácticamente llenas a estaciones que estén vacías [3].

Hay una serie de factores que influyen notoriamente sobre la demanda, los cuales son: hora del día, día de la semana y el factor meteorológico. [5]. Por tanto, es interesante realizar estudios teniendo en cuenta estos factores para una mejor gestión. [4].

Para recabar estas casuísticas, la mayoría de estaciones disponen de mecanismos de recolección de datos en tiempo real, generando grandes cantidades de datos que debemos analizar.

Gracias a estos mecanismos disponemos de un corpus de datos recopilado durante aproximadamente 7 meses, desde el 2014 al 2015 que contiene datos del uso de las bicicletas en 27 ciudades de 11 diferentes países [6]. Estos datos están acompañados de datos meteorológicos en las mismas ciudades en el mismo transcurso de tiempo. En una sección posterior se explicará con detalle los datos disponibles.

Se sabe que la corteza visual ocupa más del 50 % de nuestro cerebro, lo cual muestra la gran importancia que tiene el sentido de la vista en los seres humanos [10]. Razón por la cual consideramos conveniente introducir la utilización de gráficos para la presentación y análisis de los datos.

En este trabajo final de grado, utilizamos, transformamos y adaptamos los datos recopilados para el desarrollo de una herramienta web de análisis del uso de bicicletas en ciudades, siguiendo un modelo de diseño muy visual e intuitivo. En todo momento se relacionan los datos con el factor meteorológico puesto que, tal y como hemos visto, es uno de los factores más influyentes en el análisis de la demanda [4]. Se podrán observar patrones de uso, información general de la ciudad y de la estación escogida, influencias del factor meteorológico en la demanda, perfiles diarios y semanales, comparación de perfiles, así como el estado real de las estaciones.

## 1.1 Motivación

---

A cada instante de tiempo se están generando millones de datos que debemos gestionar. Es aquí donde entra con gran fuerza el mundo de la inteligencia artificial y, junto a ella, el análisis y visualización de datos. Lo que antes parecía algo futurista es ahora el presente.

Durante el último año del grado, se ha realizado un intercambio académico en Napier University, donde se ha cursado un módulo relacionado con el análisis de datos. Se han estudiado las técnicas y herramientas más utilizadas dentro del campo. No obstante, no se ha profundizado mucho en el tema. Es por ello, que surge un interés personal en seguir estudiando estos conceptos y así ampliar el conocimiento del campo.

Además, al ver la tendencia actual de llevar al entorno web, el mundo de las aplicaciones, nace una motivación personal de aprender alguna de las tecnologías más influyentes en este sector. Para así, poder sinergiar ambos campos y desarrollar herramientas web interesantes.

Todo ello, unido a una gran preocupación por el futuro de nuestro planeta, y con el objetivo de crear un mundo más sostenible, surge la idea de este proyecto.

Por una parte, favorecemos al usuario a que pueda visualizar de una forma interactiva el uso de las bicicletas en ciudades, generando así, un interés a la hora de utilizar este tipo de transporte. Utilizando la herramienta podrá analizar la zona y ver que estaciones están más y menos demandadas, pudiendo adaptar sus rutas utilizando estaciones con menos demanda que le asegurarán la mayoría de



veces la disponibilidad de bicis y stands. Por tanto, podrá optimizar su tiempo personal al máximo

Por otra parte, para poder descubrir, gracias al análisis de datos, patrones de uso que pueden ayudar a la optimización del sistema, dando como fruto sistemas más eficaces y atractivos para el usuario final. Utilidad que es interesante para las empresas de gestión de bicicletas.

## 1.2 Objetivos

---

El objetivo principal de este trabajo es la visualización de los datos generados por los sistemas de bicicletas compartidas. Para ello, se desarrollara una herramienta web donde tanto los usuarios del sistema como las empresas podrán visualizar los datos del uso de bicicletas. Pudiendo así, descubrir patrones de uso para poder hacer frente a los problemas de demanda. La herramienta seguirá un diseño donde toda la información estará contenida a modo de visualizaciones, las cuales, serán accesibles a través de un menú lateral dividido en cuatro secciones principales: información general de la ciudad, información de las estaciones, comparación de perfiles y tiempo real.

Para poder cumplir el objetivo principal, dividimos el proyecto en objetivos secundarios más alcanzables y medibles como son:

- Interpretar los datos disponibles: Uso bicicletas y datos meteorológicos.
- Estudiar el lenguaje de programación R y la librería Shiny.
- Estudiar las diferentes técnicas de visualización más efectivas.
- Estudiar los conceptos básicos de diseño web.
- Estudiar otros conceptos clave para el desarrollo de la herramienta: leaflet, ggplot2, API's.
- Diseñar la base de la herramienta web siguiendo un diseño "Dashboard" separando los cuatro apartados nombrados anteriormente.
- Transformar los datos para el correcto funcionamiento de la herramienta.
- Obtener los datos de interés del uso de las bicicletas.
- Implementar de las diferentes visualizaciones en cada uno de los apartados.

Gracias al desarrollo de todos estos objetivos secundarios, se alcanzará el objetivo principal del proyecto: Disponer de una herramienta web para la visualización de datos en ciudades.

## 1.3 Estructura de la memoria

---

La memoria empieza con el correspondiente resumen en los tres idiomas principales de la universidad, que son las lenguas co-oficiales de la comunidad autónoma más la lengua extranjera (inglés).

A continuación, presentamos el índice general con el contenido de toda la memoria. El cuerpo de la memoria está dividido en siete capítulos, los cuales se detallan a continuación:

- **Introducción:** Se realiza una introducción al tema propuesto, poniendo en contexto al lector sobre el campo en el que se desarrolla este proyecto. Se explica la motivación personal y profesional junto con los objetivos que se pretenden alcanzar. Con este apartado se concluye el primer punto del índice donde se explica la estructura de la memoria.
- **Estado del arte:** En este capítulo se explica detalladamente el funcionamiento de las aplicaciones más influyentes dentro del campo de estudio donde se encuentran una serie de mejoras que justifican el desarrollo de este proyecto.
- **Análisis del problema:** En este capítulo se analiza en profundidad la herramienta a desarrollar, determinando los requisitos del sistema. Una vez obtenidos los requisitos, se realiza una exploración de posibles soluciones, recalando ventajas e inconvenientes de cada una de ellas.
- **Diseño de la aplicación:** Capítulo donde se detalla cada una de las fases del diseño de la herramienta. Empezando por la arquitectura del software que explica la arquitectura que se ha seguido y la organización de los ficheros. El apartado sigue con una explicación de cómo están los datos almacenados en el sistema. Se continúa explicando detalladamente el diseño de la interfaz del usuario y se finaliza con una breve explicación de todas las tecnologías utilizadas para el desarrollo de este proyecto.
- **Implementación:** En este punto se explica con un nivel medio de abstracción la funcionalidad de la herramienta detallando como se han implementado cada una de las utilidades.
- **Ejemplo de análisis de datos:** Capítulo donde se hace un análisis de los datos de una de las ciudades disponibles, utilizando la herramienta desarrollada en este proyecto.
- **Conclusión:** Tras todas las fases de este proyecto se explican cuales han sido las conclusiones obtenidas y las posibles ampliaciones que consideramos oportunas.
- **Bibliografía :** Referencias que abarcan todos los recursos utilizados para el desarrollo de este proyecto.

---

---

# CAPÍTULO 2

## Estado del arte

---

Dentro del mundo de los sistemas de compartición de bicicletas existen diferentes aplicaciones. La mayoría de ellas sobre plataformas móviles: Android e iOS. Un gran porcentaje de estas aplicaciones están enfocadas a la productividad del usuario a la hora de consultar el estado de las estaciones. Otras muchas, tienen el objetivo de poder crear rutas dentro de las ciudades dependiendo de la proximidad de las estaciones.

En cambio, en nuestro proyecto nos desviamos de lo expuesto en el párrafo anterior y nos centramos en un objetivo central que es la visualización, análisis y interpretación de los datos que se generan diariamente en cada una de las estaciones pudiendo visualizar patrones de uso. Dentro de este campo, el número de aplicaciones se reduce considerablemente que deriva en un cambio de la plataforma móvil a la aplicación web.

### 2.1 Soluciones actuales

---

Dentro del mundo de la visualización existen diferentes soluciones. A continuación, se describen algunos de los ejemplos más desarrollados y populares:

- **O'Brien Bike Share Map:** Herramienta web que muestra la localización de estaciones en más de 300 ciudades alrededor del mundo. Las estaciones están representadas mediante círculos en el mapa, utilizando el color y tamaño del círculo para representar gráficamente la demanda actual de cada una de las estaciones. Dispone de un apartado donde se puede realizar un análisis global sobre el número total de bicicletas, número de bicicletas en uso y estación más y menos concurrida.

Además, se puede realizar un análisis de una ciudad en concreto: número de bicis activas e inactivas, factores de carga y análisis espacial de la ciudad. O bien un análisis más específico de cada una de las estaciones donde se puede ver la demanda en las últimas 24 horas. La herramienta se limita al análisis del uso de las bicicletas a partir de los datos en tiempo real de las estaciones. No tiene en cuenta otros factores como puede ser el meteorológico.

Por último, los datos que maneja son los recopilados en las últimas 24 horas, por lo que no se puede realizar un análisis a largo tiempo [12].

URL: <http://bikes.oobrien.com/#zoom=3&lon=60&lat=40>

- **Citybikes:** Esta herramienta reúne el mayor número de datos sobre el uso de las bicicletas, con más de 400 ciudades alrededor de todo el mundo. Dispone de un mapa interactivo donde se puede ver en tiempo real la demanda de cada una de las estaciones.

El objetivo principal de esta herramienta no es la visualización, ya que solamente se pueden ver las bicicletas alquiladas y las disponibles. No obstante, es la fuente de información de muchísimas otras aplicaciones, ya que dispone de una API de la cual se puede obtener la información de cada una de las estaciones.[13]

URL: <https://citybik.es/>

- **Bike Visualization:** Esta herramienta contempla 25 ciudades, la mayoría de ellas en Estados Unidos pero también se incluye Taiwán, Canadá, Reino Unido, Alemania, Noruega y México. En contraposición a la herramienta anterior, Bike Visualization está basada sobre datos recopilados durante un periodo de tiempo concreto. La fecha de estos datos varía dependiendo de la ciudad y la mayoría de ellos, se encuentran entre 2016 y 2018.

La herramienta no utiliza un mapa para la ubicación de las estaciones, se seleccionan las ciudades mediante un botón y las estaciones a través de un selector. Se puede navegar por cuatro diferentes apartados: sistema, actividad, viajes diarios y análisis de estaciones.

En el apartado sistema, se puede visualizar la información general almacenada diaria o mensualmente. Disponiendo de gráficos para el análisis del uso diario de las bicicletas, número de viajes semanales, número de viajes por hora, duración de los viajes, estaciones y rutas más populares.

También cabe destacar el uso de datos meteorológicos en el uso diario de las bicicletas, al contrario que las herramientas anteriores, en este caso sí que se muestra la diferencia del uso que hay entre días soleados y lluviosos.

El apartado actividad se centra en la actividad de cada estación sobre un mapa animado, pudiendo seleccionar un periodo de tiempo en concreto.

Así mismo, en viajes diarios se puede seleccionar un día y una hora y ver exactamente el número de viajes en cada una de las estaciones.

Por último, en el apartado análisis de actividad se puede seleccionar una estación y una fecha en concreto y visualizar tanto la actividad como los alquileres y devoluciones de las bicicletas en cada hora.

Como resultado, esta herramienta nos ha servido de inspiración para el desarrollo de este trabajo, ya que es la más completa hasta el momento.[14]

URL: <https://www.visualization.bike/>

---

## 2.2 Propuesta

---

Consideremos ahora, tras el análisis de cada una de las aplicaciones más influyentes en el campo de estudio, una serie de mejoras a implementar en nuestro proyecto.

Ninguna de las soluciones anteriores tiene en cuenta el factor meteorológico en todas sus facetas. Solamente en **Bike Visualization**[14] se puede observar que la demanda varía según si el día es lluvioso o soleado.

En nuestro caso, la diferencia reside en que disponemos de datos detallados del estado meteorológico en el mismo periodo de tiempo que la recopilación de los datos del uso de las bicicletas los cuales, se encuentran factores muy influyentes como pueden ser la temperatura, lluvia y viento [5]. Puesto que, sabemos que estos datos son relevantes a la hora del análisis, proponemos como mejora, integrarlos en nuestra herramienta. Pudiendo ver la relación existente entre el factor meteorológico y la demanda, dando así, visualizaciones más detalladas y completa en cuanto a información se refiere, con el fin de interpretar mejor los datos disponibles por parte del usuario y usuarias.

Dicho lo anterior, sabemos que existe un fuerte componente de uso periódico diario y semanal [8]. En contraste con la inexistencia en las herramientas analizadas anteriormente. Por tanto, consideramos interesante poder visualizarlo tanto en estaciones como en ciudades.

En particular, hemos visto que en ninguna de las soluciones podemos comparar el uso entre distintas ciudades o estaciones. Creemos que es interesante poder visualizarlas en conjunto para poder comparar factores y extraer conclusiones interesantes. Para ello, proponemos una utilidad donde se podrá elegir dos estaciones o ciudades y visualizar las diferencias entre los perfiles periódicos diarios y/o semanales.

Aún así, queremos seguir manteniendo utilidades que ya ofrecen las otras aplicaciones, como es el análisis en tiempo real. Pudiendo seleccionar una estación y visualizar su estado actual.

En consecuencia, nos gustaría añadir como mejora la distinción visual entre estaciones con demanda alta, normal y baja y la notificación al usuario cuando una estación está vacía ya que, consideramos que la unificación de datos pasados junto a los presentes enriquecen la herramienta y abren puertas a posibles mejoras como podría ser la predicción de la demanda futura.



---

---

## CAPÍTULO 3

# Análisis del problema

---

Tras el estudio del arte se procede a realizar un análisis en profundidad de la propuesta de este proyecto.

En primer lugar, se analizará la idea en profundidad para obtener requisitos funcionales y no funcionales.

En segundo lugar, se analizarán las posibles tecnologías que nos pueden ayudar al desarrollo de este proyecto, estudiando las ventajas e inconvenientes de cada una de ellas, obteniendo como resultado la opción que más se ajusta a la propuesta de este proyecto.

Por último, se analizarán con detalle los datos disponibles, explicando como se han obtenido y preparado para el desarrollo de este proyecto.

### 3.1 Análisis de requisitos

---

En este apartado se detallan los requisitos de la herramienta. Cada uno de ellos cuenta con un identificador único, un tipo, un título, una descripción y el apartado de la herramienta al que corresponde. El tipo de requisito puede ser funcional o no funcional y los apartados son los siguientes: general, diseño página, información ciudad, información estaciones, comparación perfiles y tiempo real.

#### Requisitos funcionales:

<b>Identificador</b>	REQF_01
<b>Tipo</b>	Funcional
<b>Título</b>	Diseño base herramienta
<b>Descripción</b>	La herramienta seguirá un diseño dashboard con un menú lateral izquierdo desde el cual se podrá acceder a las diferentes secciones.
<b>Apartado</b>	Diseño página

**Tabla 3.1:** REQF1: Diseño base de la herramienta

<b>Identificador</b>	REQF_02
<b>Tipo</b>	Funcional
<b>Título</b>	Menú lateral izquierdo
<b>Descripción</b>	El usuario podrá elegir una de las 27 ciudades mediante un selector. Podrá moverse por las diferentes secciones de la aplicación: información ciudad, información estaciones, comparación perfiles y tiempo real. Dispondrá de un enlace al código fuente de la aplicación
<b>Apartado</b>	Diseño página

**Tabla 3.2:** REQF2: Menú lateral izquierdo

<b>Identificador</b>	REQF_03
<b>Tipo</b>	Funcional
<b>Título</b>	Información genérica ciudad
<b>Descripción</b>	Cuando una ciudad este seleccionada, el usuario podrá visualizar información genérica como: ubicación ciudad en mapa, población y número de estaciones.
<b>Apartado</b>	Información ciudad

**Tabla 3.3:** REQF3: Información de la ciudad seleccionada

<b>Identificador</b>	REQF_04
<b>Tipo</b>	Funcional
<b>Título</b>	Ranking ciudad
<b>Descripción</b>	El usuario tiene que poder visualizar en qué posición está la ciudad en el ranking de uso. Siendo 1 la ciudad que más utiliza la bicicleta y 27 la que menos.
<b>Apartado</b>	Información de la ciudad

**Tabla 3.4:** REQF4: Ranking uso ciudad

<b>Identificador</b>	REQF_05
<b>Tipo</b>	Funcional
<b>Título</b>	Gráficos meteorológicos
<b>Descripción</b>	El usuario podrá visualizar la temperatura, lluvia y viento mediante el uso de gráficos dentro del período de tiempo seleccionado.
<b>Apartado</b>	Información de la ciudad

**Tabla 3.5:** REQF5: Gráficos meteorológicos ciudad



<b>Identificador</b>	REQF_06
<b>Tipo</b>	Funcional
<b>Título</b>	Gráfico demanda y tiempo
<b>Descripción</b>	El usuario podrá visualizar el uso de las bicicletas en la ciudad junto a los datos meteorológicos. Tendrá la posibilidad de seleccionar el modo de visualización: diario o mensual. Así como, los datos de interés meteorológico: descripción tiempo y temperatura y podrá a la vez, distinguir entre días de la semana o fin de semana.
<b>Apartado</b>	Información de la ciudad

**Tabla 3.6:** REQF6: Visualización uso de bicicleta y tiempo

<b>Identificador</b>	REQF_7
<b>Tipo</b>	Funcional
<b>Título</b>	Perfiles periódicos generales ciudad
<b>Descripción</b>	El usuario podrá visualizar el perfil periódico diario y semanal de una ciudad.
<b>Apartado</b>	Información de la ciudad

**Tabla 3.7:** REQF7: Perfiles diarios y semanales ciudad

<b>Identificador</b>	REQF_08
<b>Tipo</b>	Funcional
<b>Título</b>	Comparar perfiles demanda
<b>Descripción</b>	El usuario podrá seleccionar dos ciudades y comparar sus perfiles periódicos. También, tendrá la posibilidad de comparar estaciones eligiendo dos ciudades y dos estaciones.
<b>Apartado</b>	Comparación perfiles de estaciones y ciudades

**Tabla 3.8:** REQF8: Comparar perfiles estaciones y ciudades

<b>Identificador</b>	REQF_9
<b>Tipo</b>	Funcional
<b>Título</b>	Mapa con estaciones
<b>Descripción</b>	El usuario dispondrá de un mapa con todas la estaciones. Cada estación, tendrá un color diferente según demanda, siendo el rojo una estación muy demandada, naranja con demanda media y verde con poca demanda.
<b>Apartado</b>	Información estaciones

**Tabla 3.9:** REQF9: Mapa de estaciones según demanda

<b>Identificador</b>	REQF_10
<b>Tipo</b>	Funcional
<b>Título</b>	Selección y visualización de estaciones
<b>Descripción</b>	Al seleccionar una estación del mapa, se mostrará la siguiente información: estación con más demanda, estación con menos demanda, número de stands y servicios de los que dispone la estación (banco y/o bonos).
<b>Apartado</b>	Información estaciones

**Tabla 3.10:** REQF10: Información detallada de la estación

<b>Identificador</b>	REQF_11
<b>Tipo</b>	Funcional
<b>Título</b>	Gráfico demanda estación
<b>Descripción</b>	El usuario visualizará la demanda de la estación seleccionada pudiendo seleccionar el modo de visualización: diario o semanal y el día deseado dentro del rango de los datos.
<b>Apartado</b>	Información estaciones

**Tabla 3.11:** REQF11: Gráfico demanda estación

<b>Identificador</b>	REQF_12
<b>Tipo</b>	Funcional
<b>Título</b>	Perfil periódico estación
<b>Descripción</b>	El usuario podrá visualizar el perfil periódico diario y semanal de la estación seleccionada.
<b>Apartado</b>	Información estaciones

**Tabla 3.12:** REQF12: Perfil semanal y diario estación

<b>Identificador</b>	REQF_13
<b>Tipo</b>	Funcional
<b>Título</b>	Visualizar estaciones en tiempo real
<b>Descripción</b>	Existirá un mapa el cual se cargará con los datos recibidos en la llamada a la API. El mapa, coloreará los iconos según las bicicletas disponibles de la estación. Si una estación tiene más de 10 bicicletas disponibles, se coloreará en verde. Si tiene menos de 10 pero más de 5, se coloreará en naranja y si tiene menos de 5, en rojo. El mapa se actualizará cada dos minutos.
<b>Apartado</b>	Tiempo real

**Tabla 3.13:** REQF13: Mapa de estaciones en tiempo real

<b>Identificador</b>	REQF_14
<b>Tipo</b>	Funcional
<b>Título</b>	Información en tiempo real estación seleccionada
<b>Descripción</b>	Cuando el usuario seleccione una estación se mostrará información de interés: número de stands disponibles, número de bicicletas disponibles y número de stands vacíos.
<b>Apartado</b>	Tiempo real

**Tabla 3.14:** REQF14: Información estación en tiempo real

### Requisitos no funcionales:

<b>Identificador</b>	REQNF_1
<b>Tipo</b>	No Funcional
<b>Título</b>	Número de usuarios
<b>Descripción</b>	La herramienta debe ser capaz de funcionar correctamente con 50 usuarios.
<b>Apartado</b>	General

**Tabla 3.15:** REQNF1: Número de usuarios

<b>Identificador</b>	REQNF_2
<b>Tipo</b>	No Funcional
<b>Título</b>	Tiempo respuesta API
<b>Descripción</b>	Los datos de la API deben ser obtenidos en menos de 3 segundos, por lo que se requiere una buena conexión a Internet.
<b>Apartado</b>	General

**Tabla 3.16:** REQNF2: Obtención datos API

<b>Identificador</b>	REQNF_3
<b>Tipo</b>	No Funcional
<b>Título</b>	Tiempo de carga de las visualizaciones
<b>Descripción</b>	Los datos deben ser gestionados de la mejor manera para que las visualizaciones sean cargadas rápidamente. El sistema debe responder en menos de 1 segundo.
<b>Apartado</b>	General

**Tabla 3.17:** REQNF3: Carga de visualizaciones

<b>Identificador</b>	REQNF_4
<b>Tipo</b>	No Funcional
<b>Título</b>	Tiempo de aprendizaje
<b>Descripción</b>	El tiempo de aprendizaje de la herramienta debe ser inferior a 1 hora. Para ello se mostrará un ejemplo de uso en la documentación.
<b>Apartado</b>	General

**Tabla 3.18:** REQNF4: Tiempo máximo de aprendizaje

<b>Identificador</b>	REQNF_5
<b>Tipo</b>	No Funcional
<b>Título</b>	Interfaces gráficas
<b>Descripción</b>	El sistema estará formado por interfaces gráficas bien diseñadas. Aportando así, un sistema intuitivo y fácil de utilizar.
<b>Apartado</b>	General

**Tabla 3.19:** REQNF5: Interfaces gráficas

<b>Identificador</b>	REQNF_6
<b>Tipo</b>	No Funcional
<b>Título</b>	Navegadores disponibles
<b>Descripción</b>	La herramienta debe poder se accedida desde los dos navegadores más influyentes en el mercado: Google Chrome y Mozilla Firefox.
<b>Apartado</b>	General

**Tabla 3.20:** REQNF6: Navegadores disponibles

## 3.2 Análisis de tecnologías

En este apartado se comparan las diferentes tecnologías analizadas para el desarrollo de este proyecto centrándonos en las más influyentes dentro del mundo del análisis de datos y la visualización.

- Python + Dash:** Python es uno de los lenguajes más populares dentro del mundo del análisis de datos y inteligencia artificial. Dispone de muchísimas librerías para el manejo de datos. Teniendo disponible funciones para cálculos matemáticos y estadísticos. Es un lenguaje fácil de aprender y con gran crecimiento en los últimos años [18].

Dash es un framework de Python que permite crear aplicaciones para la visualización de datos. Proporciona una manera rápida y eficaz para la creación de herramientas web sin la necesidad de aprender todos los conceptos necesarios dentro del mundo web.

- **R + Shiny:** R es otro lenguaje muy popular dentro del análisis de datos. Dispone también de muchísimas librerías para realizar cálculos matemáticos y estadísticos. Está más enfocado hacia el análisis de datos y visualización. Dispone de librerías muy potentes de creación de gráficos.

Shiny es un paquete para R que permite crear aplicaciones web interactivas. Abstrayendo los conceptos de programación web, proporcionando así, una manera muy rápida y eficaz de creación de contenido. Está muy enfocado en la reactividad por lo que proporciona mecanismos fáciles de utilizar para crear páginas web interactivas.

Si comparamos cada tecnología, podemos ver cuáles son sus principales diferencias y cuál es la que mejor se adapta a nuestras necesidades.

Primeramente, vamos a centrarnos en las bases de cada uno de los lenguajes. R es un lenguaje que ha sido desarrollado para el análisis estadístico. Python, tiene múltiples salidas, es muy útil para el estudio estadístico pero puede ser utilizado para otros fines y ambos, son los más utilizados para el análisis y exploración de datos [22].

Si nos centramos en el mundo de la visualización, R dispone de paquetes muy interesantes y extendidos para la creación de gráficos. Por el contrario, Python, a pesar de tener diferentes paquetes de visualización, no dispone de la misma potencia que R [22].

Ambos tienen ventajas comunes, son *open source*, disponen de herramientas y paquetes para el análisis de datos. Están apoyados por grandes comunidades, por lo que hay muchísima documentación al respecto [22].

En cuanto a Shiny y Dash, ambos son muy similares y proporcionan prácticamente las mismas funcionalidades. Disponen de mecanismos de reactividad, facilitando la creación de contenido web interactivo.

Como el principal objetivo de este proyecto es la visualización de datos del uso de las bicicletas, tras analizar detalladamente cada lenguaje, hemos llegado a la conclusión de que en este caso R, proporciona mejores librerías para la creación de gráficos. Necesitamos un lenguaje que disponga de herramientas potentes de visualización. En este caso, R se adapta perfectamente a nuestras necesidades. Además, utilizando el paquete Shiny podemos cubrir todos nuestros requisitos, pudiendo crear una herramienta web totalmente interactiva y reactiva.

### 3.3 Datos disponibles

---

Disponemos de un dataset con información meteorológica y del uso de bicicletas recavado en 27 ciudades dentro de 11 diferentes países. Exactamente, disponemos de datos de 3.584 estaciones, de las cuales 2.295 están ubicadas en ciudades de Francia. Las estaciones tienen diferentes tamaños, desde 10 stands por estación, hasta 50 y con una media de 20 por estación. En el mismo periodo de tiempo, se han almacenado datos meteorológicos dentro en cada una de las ciudades [6]. El dataset está públicamente accesible en <http://dmip.webs.upv.es/BikeSharingDemand/>.

Para la recolección de estos datos se han utilizado los siguientes servicios:

- **JCDcaux:** Líder en servicios de alquiler de bicicletas. Desde 2013 están apostando por los datos abiertos, ofreciendo una API para el acceso a los datos que se generan en cada una de las estaciones. Se pueden obtener datos estáticos y dinámicos. Los datos estáticos se pueden obtener a través de la API o descargando un fichero de texto. Para poder acceder a los datos dinámicos, se necesita un registro y obtención de una API Key, sin ningún coste asociado. La API actualiza los datos cada 2 minutos y pueden ser obtenidos en formato JSON mediante llamadas GET, pasando como argumento la API key. [19].
- **OpenWeatherMap:** Se han obtenido los datos meteorológicos haciendo uso de la API de OpenWeatherMap. Este servicio ofrece un plan sin coste, donde se pueden realizar hasta 60 llamadas a la API por minuto. Se pueden obtener los datos meteorológicos, alertas y predicciones en tiempo real [20].

El dataset ha sido recolectado durante 7 meses, desde el 26 de Septiembre del 2014 hasta el 25 de Abril del 2015. Los datos han sido recopilados cada 2 minutos. Posteriormente, han sido agregados por horas para reducir el volumen de datos. Desde las 0 hasta las 23 horas de cada día.

Los datos que recogemos de cada estación son los siguientes:

- **Datos estáticos:** número de estación, nombre, dirección, localización geográfica, ciudad, número de stands y si hay servicio de banca y bonos.
- **Datos dinámicos:** timestamp de la última actualización, número bicis disponibles, número de stands vacíos, estado de la estación.

Tras la agregación de los datos dinámicos por horas se han calculado los siguientes atributos: total de incrementos, total de decrementos, mediana bicicletas, media bicicletas, número de bicis en la última hora y probabilidad de que esté vacía la estación.

Estos datos agregados son utilizados en nuestra herramienta web, ya que, nos proporcionan la información necesaria para poder analizar la demanda.

En cuanto a los datos meteorológicos, son todos dinámicos y accesibles mediante la API.

Disponemos de los siguientes: hora, fecha, coordenadas, humedad, presión, temperatura, temperatura máxima, temperatura mínima, hora que amanece, hora que anochece, viento, lluvia, nieve, fecha y descripción del tiempo.

La recopilación de datos ha sido en periodos de 15 minutos. No obstante, se han agregado posteriormente por horas.

Al igual que en los datos del uso de bicicletas, utilizamos los datos agregados, puesto que, nos proporcionan la información necesaria para la implementación de la herramienta.

---

# CAPÍTULO 4

## Diseño de la aplicación

---

En este apartado nos centramos en como vamos a diseñar nuestra herramienta web.

Primero, se analizará la arquitectura de software a seguir. Seguidamente, se explicará como están almacenados los datos en nuestra herramienta.

Posteriormente, se explicará en detalle el diseño de la interfaz del usuario terminando la sección con una explicación de cada una de las tecnologías utilizadas en el desarrollo de este proyecto.

### 4.1 Arquitectura del software

---

En nuestro proyecto vamos a seguir una arquitectura cliente/servidor. En el siguiente diagrama, Figura 4.1, podemos visualizar como estará distribuida la aplicación.

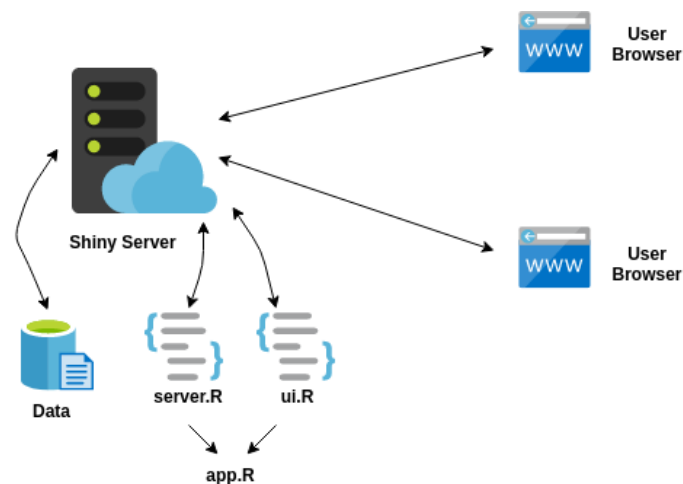


Figura 4.1: Arquitectura del proyecto

Una aplicación Shiny básica está formada principalmente por dos componentes: la parte interfaz de usuario (UI) **ui.R** y la parte servidora **server.R**. El fichero UI controla la apariencia de la web, el fichero server la lógica. Ambos se unifican dentro del fichero **app.R** donde se hace una llamada a la función `shinyApp` pasando como argumento la parte de UI y la parte de servidor. [15].

Cuando un programa crece, hay que buscar soluciones. No es viable contener todo el código dentro de dos ficheros. En consecuencia, es imprescindible tener una buena organización para que sea fácil de mantener.

Para tener una mejor distribución del código vamos a separar ambas partes en dos directorios **server** y **ui**. Siguiendo una aproximación que consiste en dividir el código dependiendo de la lógica de nuestra aplicación [16].

En nuestro caso, tal y como se ha comentado en el apartado del análisis de requisitos, la aplicación estará dividida principalmente en cuatro secciones: información de la ciudad, información de las estaciones, comparación de perfiles y tiempo real. Por lo que vamos a crear un fichero UI y un fichero server para cada una de estas secciones.

De esta forma, tendremos una mejor organización de nuestro código, que nos facilitará posteriormente el mantenimiento de la aplicación. Adicionalmente, se creará un fichero en el directorio servidor que contendrá las funciones necesarias para el procesamiento de datos: **functions.R**.

El siguiente diagrama, Figura 4.2 muestra la organización de directorios y ficheros que seguirá nuestra herramienta web.

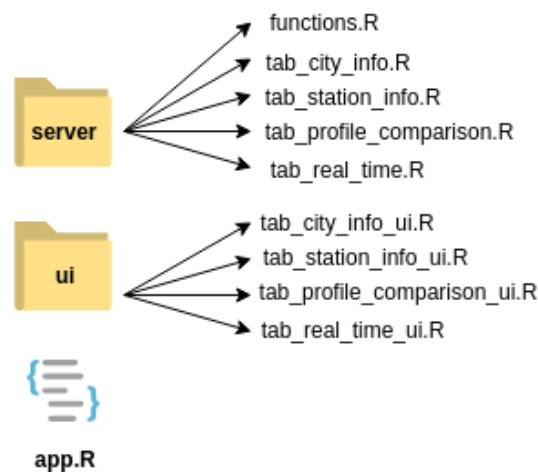


Figura 4.2: Organización directorios y ficheros en la herramienta

Como se puede observar en Figura 4.2 se separa completamente la parte del servidor y la interfaz de usuario. Dentro del fichero **app.R** se referenciarán cada uno de estos ficheros para la correcta ejecución de la aplicación.

## 4.2 Almacenamiento de los datos

En nuestro proyecto, solamente accedemos a los datos en modo lectura, por lo que hemos optado por no crear ninguna base de datos y leer los datos directamente de los ficheros CSV obtenidos.

Para una mejor gestión y acceso a los datos, se han estructurado dentro de diferentes directorios por una parte, los datos del uso de las bicicletas: **bikes\_aggregated** y por otra, los meteorológicos: **weather\_aggregated**.



En los datos meteorológicos, cada fichero CSV tiene el nombre de la ciudad asociada. En los datos del uso de las bicicletas, se ha creado un subdirectorio para cada una de las ciudades y estaciones donde el nombre del directorio y del fichero CSV siguen el formato *Ciudad:Estación*.

A continuación, podemos ver un ejemplo aclaratorio de como queda estructurado en el sistema de ficheros:

```
datasets
├── bikes_aggregated
│   ├── Dublin:1
│   │   └── Dublin:1.csv
│   ├── Dublin:2
│   │   └── Dublin:2.csv
│   └── ...
```

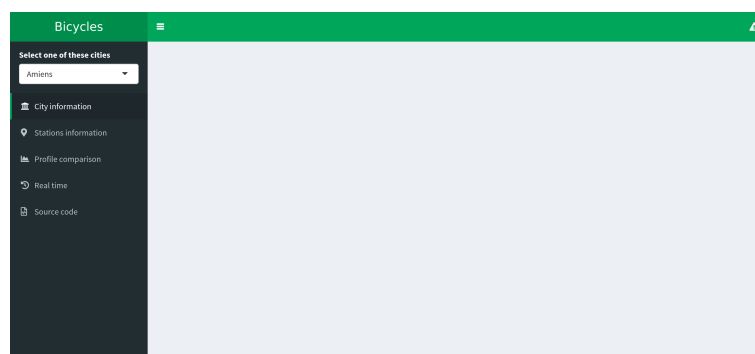
Los datos meteorológicos se estructuran por nombre de ciudad:

```
datasets
├── weather_aggregated
│   ├── Amiens.csv
│   ├── Dublin.csv
│   └── ...
```

## 4.3 Diseño de la interfaz

Tras el análisis de requisitos procedemos a realizar un diseño de la interfaz del usuario. Optamos por un diseño fácil e intuitivo de cara al usuario final. Por tanto, vamos a seguir un diseño donde la información quedará contenida en forma de tablón. El usuario mediante un menú lateral podrá moverse a través de las diferentes secciones de la web.

La siguiente imagen, [4.3](#), muestra la base de la herramienta web a desarrollar.

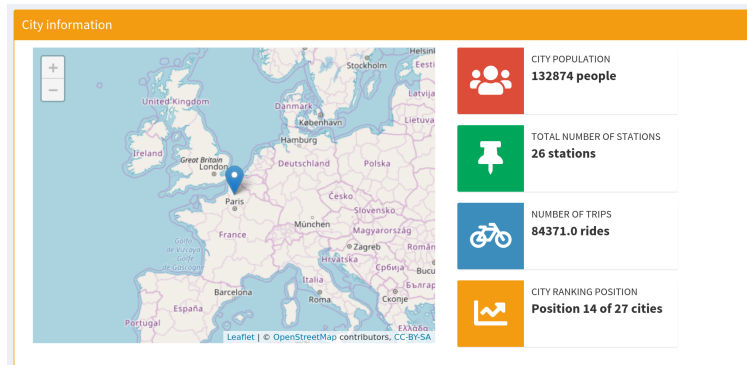


**Figura 4.3:** Diseño base de la herramienta

A continuación se detalla el contenido de cada una de las secciones de la herramienta:

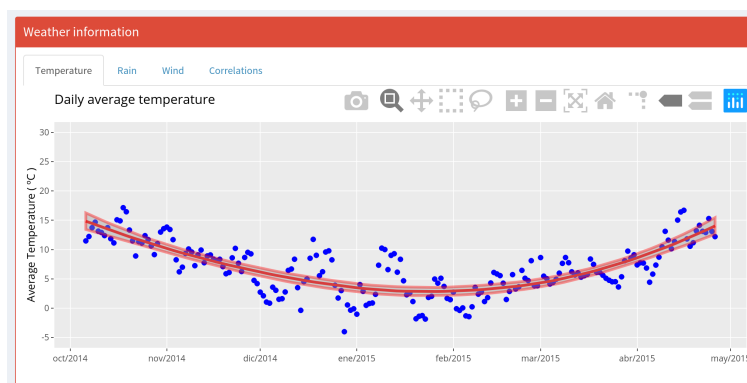
- **Información de la ciudad:** Esta sección estará formada por tres apartados donde se presentará diferente información. El primero de ellos mostrará un

mapa para poder ubicar la ciudad seleccionada. Le acompañarán una serie de cajas informativas donde se podrá visualizar información de interés general de la ciudad: número total de habitantes, número total de estaciones, número total de viajes realizados y posición en la que se encuentra la ciudad dentro del ranking de uso. En la figura 4.4 podemos ver el diseño de este apartado.



**Figura 4.4:** Apartado información general ciudades

El segundo apartado estará orientado a la visualización de los datos meteorológicos: temperatura, lluvia y viento. Cada visualización estará separada por pestañas donde el usuario podrá ir cambiando de una a otra según sus intereses. La pestaña temperatura estará formada por un gráfico lineal donde se podrá visualizar la temperatura en el transcurso del tiempo de recopilación de los datos. La pestaña lluvia dispondrá de una gráfica de dispersión donde se podrá ver la cantidad de lluvia recogida en cada uno de los días. El viento se mostrará mediante una rosa de los vientos donde se podrá visualizar la dirección y velocidad. En la pestaña correlaciones podremos ver la relación que existe entre demanda y cada una de las variables meteorológicas. La imagen 4.5 muestra el diseño para este apartado.



**Figura 4.5:** Apartado información meteorológica

Esta sección concluirá con un último apartado donde se podrá visualizar la demanda junto con el factor meteorológico y los perfiles periódicos. Para una mejor visualización, se separará el contenido en dos pestañas: vista general del uso de bicicletas y información de perfiles periódicos.

En la primera pestaña dispondremos de un panel de control donde el usuario podrá interactuar con el gráfico de demanda. Utilizando un botón radio podrá elegir si visualizar la demanda por días o meses. Mediante un grupo de check boxes podrá seleccionar si visualizar la información meteorológica: temperatura y/o descripción del día. Además, dispondrá de un check box para seleccionar si desea ocultar los fines de semana. La imagen 4.6 muestra un ejemplo de la interfaz a utilizar.

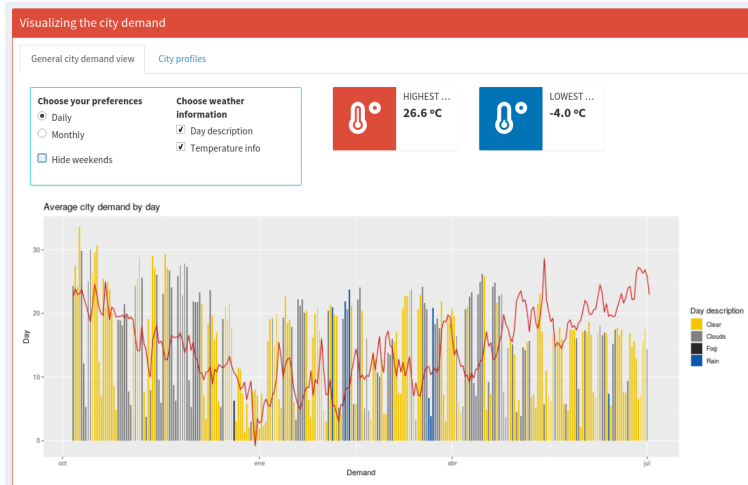


Figura 4.6: Apartado información demanda junto con meteorológica

Cabe recalcar que la escala del eje Y del gráfico muestra la demanda y no la temperatura. No obstante, consideramos interesante poder visualizar la línea de temperatura, puesto que permite ver como afecta en la demanda.

La segunda pestaña: Perfiles de la ciudad, mostrará los perfiles periódicos de la demanda de la ciudad, pudiendo seleccionar, mediante un radio botón, si visualizar el perfil diario o semanal. Para la visualización se utilizará un gráfico de barras 4.7.

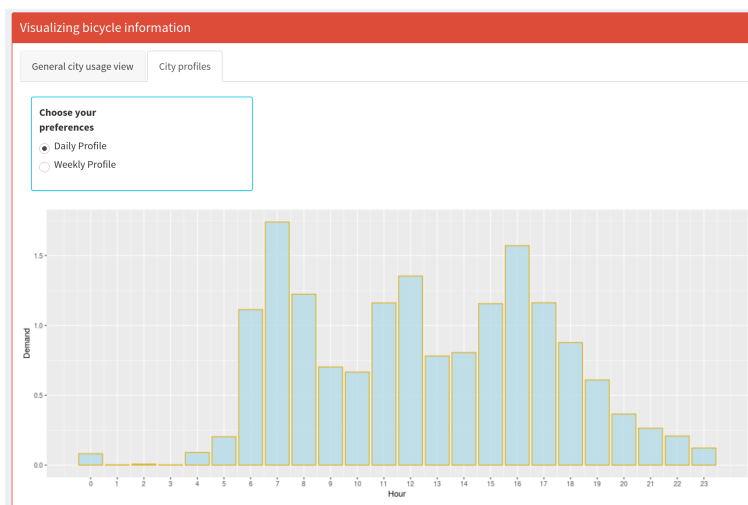


Figura 4.7: Apartado visualización perfiles ciudades

- **Información de las estaciones:** Siguiendo el mismo estilo que la sección anterior se va a seguir un diseño por apartados.

En el primero se ubicarán cada una de las estaciones en un mapa. Cada marcador representará la demanda de la estación, siendo rojo máxima demanda, naranja demanda normal y verde mínima. El usuario podrá seleccionar cualquier marcador para obtener información más detallada de la estación: número de stands, servicios disponibles (banca y/o bonos) y estación con mayor y menor demanda. Esta información estará contenida en cajas informativas.

La imagen 4.8 muestra el diseño a seguir.

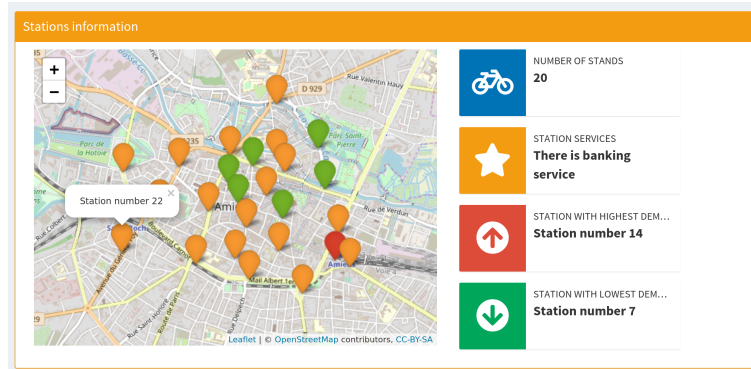


Figura 4.8: Apartado informativo de las estaciones

En el segundo se separará la visualización de la demanda y los perfiles periódicos. Para ello, se utilizará un diseño en pestañas. En la pestaña de visualización de demanda, el usuario, mediante un radio botón, podrá elegir el modo de visualización (diario o semanal), y una fecha que se podrá seleccionar haciendo uso de un selector de fechas. Este selector solo mostrará las fechas disponibles dentro del rango de tiempo de nuestros datos.

Se utilizará un gráfico de barras para la visualización de los datos, donde el eje Y será la hora de la semana o del día y el eje X la demanda. En la imagen 4.9 muestra el diseño a seguir.

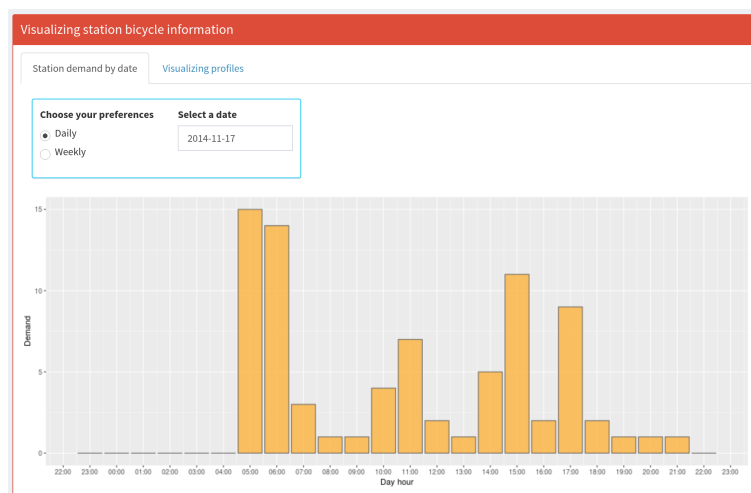


Figura 4.9: Apartado visualización demanda según fecha

En la pestaña de visualización de los perfiles de la estación el usuario dispondrá de un radio botón para elegir si ver el perfil diario o el semanal. Estos datos se mostrarán mediante un gráfico de barras 4.10.

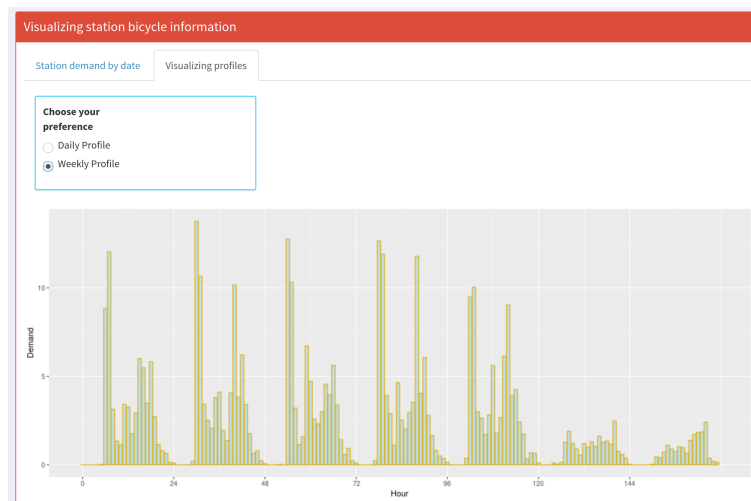


Figura 4.10: Apartado visualización perfiles estación

- **Comparando perfiles:** Esta sección estará formada por dos apartados. El primero nos permite comparar los perfiles de ciudades. En el segundo, se podrán comparar los de perfiles de estaciones.

El apartado de comparación de perfiles de ciudades dispondrá de un panel de control donde el usuario podrá seleccionar dos ciudades y el modo de visualización que desea. Para la selección de ciudades se utilizara un combo box donde estarán cargadas todas las ciudades que tenemos disponibles. Para el modo de visualización un radio botón con dos posibles opciones, vista diaria o semanal. Junto al panel de control aparecerá un gráfico lineal donde se mostrará el perfil de cada una de las ciudades. Pudiendo así, realizar una comparación visual.

La imagen 4.11 muestra el diseño de este apartado.

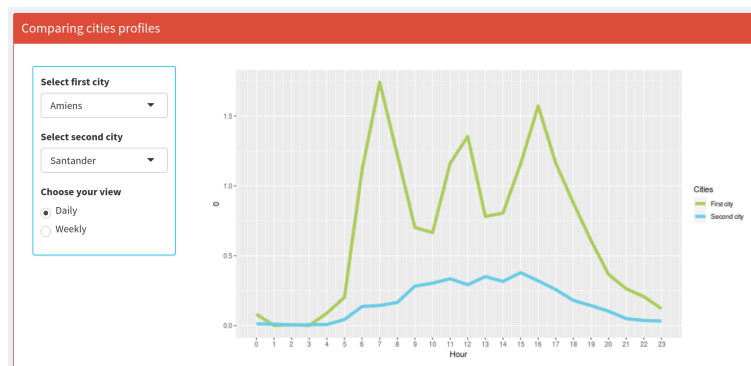


Figura 4.11: Comparación perfiles ciudades

El segundo apartado será muy similar. No obstante, en este caso el usuario dispondrá de otro combo box para la elección de estaciones. Este segundo

combo se cargará dinámicamente cuando el usuario haya seleccionado una ciudad. La imagen 4.12 muestra el diseño de la interfaz de este apartado.

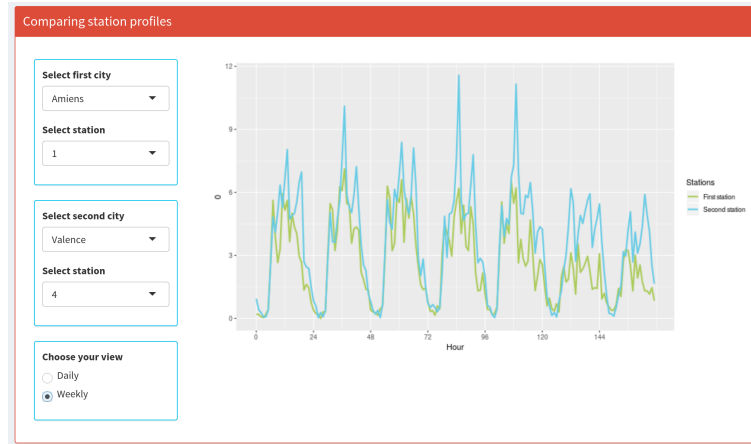


Figura 4.12: Comparación perfiles estaciones

- Tiempo real:** En esta sección se podrá visualizar el estado real de cada una de las estaciones haciendo el uso de un mapa interactivo. El número de bicicletas disponibles en cada estación quedará representado por el color del marcador de estación. Siendo rojo el color que identifica estaciones con pocas o ninguna bicicleta disponible. Naranja el color para representar estaciones con menos de diez bicicletas y verde para la representación de estaciones con más de diez bicicletas disponibles.

Cuando el usuario seleccione alguno de los marcadores, se mostrará en cajas informativas la información específica de la estación: número de stands totales, número de bicicletas disponibles y número de stands vacíos.

Por otro lado existirá un texto que informará de la fecha y hora de la última actualización del mapa.

La imagen 4.13 muestra el diseño para esta sección.

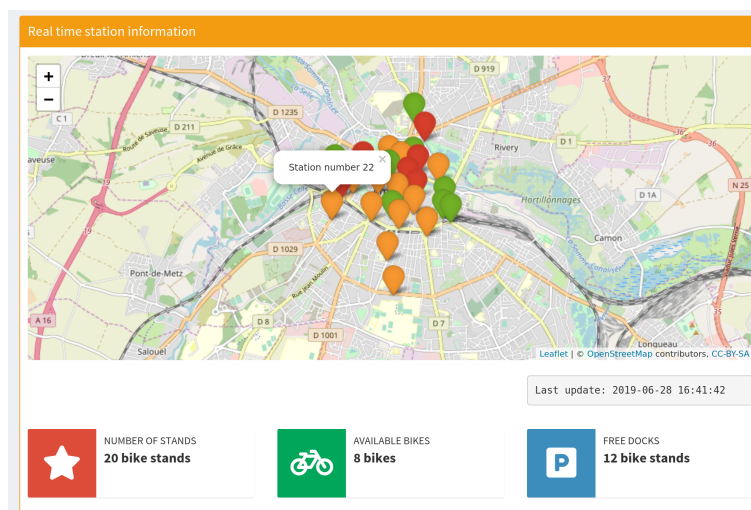


Figura 4.13: Primera caja sección ciudades

## 4.4 Tecnología utilizada

---

En este apartado se detallan cada una de las tecnologías utilizadas. Se explica en qué consisten y qué aportan al desarrollo de este proyecto.

- **Lenguaje R:** R es un entorno y lenguaje de programación con un enfoque al análisis estadístico y gráfico. Es un lenguaje con una sintaxis simple e intuitiva. Es orientado a objetos e interpretado, lo que nos facilita la ejecución de comandos en la terminal, pudiendo así realizar pruebas y manejar nuestros datos sin la necesidad de compilar el programa [17].

En este proyecto se decide utilizar este lenguaje puesto que el principal objetivo de este proyecto es la visualización de datos. Al estar completamente enfocado al análisis estadístico y gráfico disponemos de un amplio abanico de librerías que nos facilitarán mucho el desarrollo de este proyecto.

- **RStudio IDE** Entorno de desarrollo integrado para el lenguaje de programación R. Esta formado por distintas herramientas que facilitan el trabajo, entre ellas cabe destacar: la consola, el editor de sintaxis que apoya el código y las herramientas de trazado, gestión y depuración del código [?].

En cuanto a las características principales, destacar que está construido exclusivamente para el lenguaje R, por lo que proporciona utilidades de auto-completado de código, resaltado de sintaxis y sangría inteligente. Se puede ejecutar todo el código directamente desde el editor o bien seleccionar la ejecución una porción para ver los resultados.

RStudio nos facilitará el desarrollo del proyecto. Por tanto, se utilizará durante todo el proceso de implementación.

- **Librería shiny:** Tal y como se a introducido en el apartado análisis de tecnologías, vamos a utilizar la librería Shiny para el desarrollo de nuestra herramienta web. Shiny es un paquete open-source que proviene de un elegante y poderoso framework para la creación de paginas web utilizando el lenguaje de programación R. Nos permite crear aplicaciones web interactivas donde poder visualizar nuestros datos [15]. No es necesario el conocimiento previo de tecnologías web. Combina el poder computacional de R junto con la interoperatividad de las paginas web.



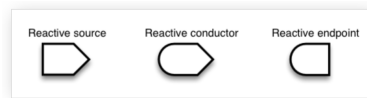
Figura 4.14: Logo librería Shiny

Además, permite la inserción de elementos dentro de la interficie del usuario, separando los elementos en entradas (inputs) y salidas (outputs). Los

elementos de entrada pueden ser: botones, checkbox, radio buttons, rangos de fechas, sliders, texto, etc. Los de salida: gráficos, tablas, mapas, imágenes, cajas informativas, etc. La interfaz de usuario se crea juntando elementos de entrada y de salida. Pudiendo organizar los elementos de diferentes formas según nuestros intereses. La obtención de los datos de los elementos de entrada así como la renderización de cada los elementos de salida serán gestionados en la parte servidora de nuestra aplicación [15].

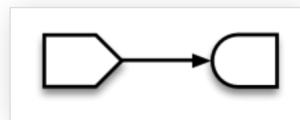
Para permitir la interactividad en las aplicaciones, Shiny implementa un modelo de programación reactiva. Diferenciando tres tipos de objetos: fuentes reactivas, conductores reactivos y puntos finales reactivos [15].

En el siguiente diagrama, 4.15, podemos ver como se representan gráficamente cada uno de estos elementos.



**Figura 4.15:** Tipos de objetos en la programación reactiva de Shiny

En nuestra herramienta no necesitamos utilizar conductores reactivos por tanto la mayoría de relaciones serán entre fuentes reactivas y finales reactivos 4.16.



**Figura 4.16:** Relación simple de reactividad

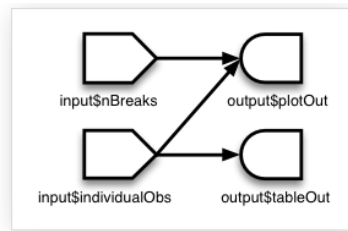
Típicamente cuando hablamos de una fuente reactiva hacemos referencia a un elemento de entrada y cuando hablamos de un punto final reactivo a los elementos de salida.

Un ejemplo podría ser un botón, elemento de entrada, que cada vez que es pulsado cambia una imagen, elemento de salida. Como podemos ver ambos elementos quedan conectados y su valor depende del otro.

Una fuente reactiva puede tener diferentes elementos de salida y viceversa. El botón podría no solo cambiar la imagen, sino también un gráfico o más elementos. O por otro lado, la imagen podría ser cambiada no solo por un botón sino por cualquier otro tipo de elemento de entrada.

En el diagrama 4.17 podemos ver un ejemplo de las diferentes relaciones que puede haber entre dos fuentes y dos puntos finales.





**Figura 4.17:** Ejemplo reactividad en Shiny

Este modelo de programación se adapta perfectamente a nuestro proyecto, tal y como se ha visto en los apartados anteriores nuestro objetivo es ofrecer una herramienta web donde los usuarios puedan interactuar utilizando elementos de entrada y visualizar los cambios en elementos de salida.

Esta representación será utilizada durante todo el proceso de implementación. En el capítulo de implementación se podrá ver la relaciones existente en cada uno de los elementos de nuestra aplicación.

- Librería ggplot2:** Paquete para la creación de visualizaciones en el lenguaje R. Fue creado por Hadley Wickham en 2005 basandose en el libro *Grammar of Graphics* para crear un esquema de visualización de datos. De ahí el nombre de ggplot: grammar of graphics plot. Desde 2005 ggplot2 a crecido hasta tal punto de convertirse en uno de los paquetes más populares dentro de R. Esta licenciado bajo GNU GPL v2. La principal diferencia respecto a otros paquetes de gráficos es el nivel de abstracción, ggplot2 permite al usuario poder añadir, borrar o intercambiar componentes dentro de los gráficos en un nivel de abstracción muy elevado, facilitando así el desarrollo de visualizaciones [21].

Este paquete será la base de nuestro proyecto. Nos permitirá crear todos los gráficos necesarios para la visualización de los datos que tenemos disponibles. Ya que es perfectamente integrable como elemento de salida dentro de la librería Shiny.

- Librería leaflet:** Leaflet es una de las librerías open-source de Javascript mas populares para la creación de mapas interactivos. Fue lanzado en 2011 por Vladimir Agafonkin. Dispone de una versión para R que es perfectamente integrable dentro de la librería Shiny. Pudiendo crear elementos de salida dentro de la interfaz del usuario de Shiny ??

Dispone de muchísimas funcionalidades, cabe destacar las más afines a nuestro proyecto: creación de marcas dentro del mapa a partir de coordenadas, visualización de información cuando el ratón pasa por encima de cada marca, posible conexión con otros elementos de la UI de la herramienta.

Todas estas funcionalidades nos permiten poder cubrir alguno de nuestros requisitos de la aplicación.



---

# CAPÍTULO 5

## Implementación

---

A continuación se describe todo el proceso de implementación de la herramienta. Para ello, vamos a seguir meticulosamente el análisis que se realizó en los apartados anteriores, implementando cada uno de los requisitos siguiendo el diseño de la interfaz propuesto.

Se muestran diagramas de la relación existente de cada uno de los elementos que componen la herramienta, ya que la principal característica de este proyecto es la reactividad.

### 5.1 Desarrollo de la interfaz de usuario

---

A continuación, se detalla como hemos implementado la interfaz de la herramienta web. Tal y como se ha visto en apartados anteriores, Shiny divide la aplicación en dos partes: la parte de la interfaz del usuario y la parte servidora. En este apartado nos centramos en la interfaz del usuario (UI).

Un dashboard se divide en tres partes: cabecera, barra lateral y cuerpo. Hemos insertado en la barra lateral un menú con las cuatro secciones que forman nuestra herramienta: información de la ciudad, información de las estaciones, tiempo real y código fuente.

Como se ha explicado en el apartado de arquitectura de software, hemos seguido una aproximación en la que se separa la lógica de la aplicación en diferentes ficheros, según cada una de las secciones del menú.

```
dashboardSidebar(  
  sidebarMenu(  
    selectInput("selected_city", "Select one of these cities", cities$NAME),  
    menuItem("City information", tabName = "city_information", icon = icon("university")),  
    menuItem("Stations information", tabName = "stations_information", icon = icon("map-marker-alt")),  
    menuItem("Real time", tabName = "real_time", icon = icon("chart-area")),  
    menuItem("Source code", icon = icon("file-code-o"), href = "https://github.com/vicennt/bicycles-use-analysis")  
  )  
)  
dashboardBody(  
  shinyjs::useShinyjs(),  
  tabItems(  
    source(file.path("ui", "tab_city_info_ui.R"), local = TRUE)$value,  
    source(file.path("ui", "tab_station_info_ui.R"), local = TRUE)$value,  
    source(file.path("ui", "tab_real_time_ui.R"), local = TRUE)$value  
  )  
)  
)
```

**Figura 5.1:** Código unificación ficheros secciones

En la figura 5.1 podemos ver la manera en la que hemos asignado los identificadores de cada ítem del menú (atributo **tabName**) y como hacemos referencia desde el `dashboardBody` a cada uno de los ficheros de la UI.

Cuando un ítem del menú es accedido, la herramienta cargará el correspondiente código que hay en el fichero, mostrando el contenido en el cuerpo de la aplicación.

Cada una de las secciones ha sido diseñada siguiendo el diseño de la interfaz propuesto en uno de los apartados anteriores. Cada sección se compone por una serie de cajas (elemento `box`), donde se insertan cada uno de los elementos de entrada o salida.

Los elementos se organizan siguiendo un diseño bootstrap, donde la página queda dividida en filas y columnas a modo de rejilla. Cada una de las filas está formada por 12 columnas. Los elementos han sido colocados siguiendo un estilo intuitivo y fácil de utilizar, adaptado para el usuario final.

Cabe destacar que hemos incluido un selector de ciudades dentro del menú. Este selector se carga dinámicamente al inicio de la aplicación a partir del fichero `ciudades`. Es una parte importante de la herramienta, puesto que tiene una relación con la mayoría de elementos de nuestra UI. Cuando el usuario cambia de ciudad en el selector, todo el contenido de la herramienta es actualizado con la información perteneciente a la ciudad seleccionada. Esto lo conseguimos gracias al modelo de programación reactiva que utiliza Shiny, mencionado anteriormente.

## 5.2 Funciones para el procesamiento de datos

---

Los datos han sido procesados para adaptarlos a la funcionalidad de la herramienta web. Para ello, se han implementado una serie de funciones que se encargan de seleccionar los datos que se necesitan para implementar toda la funcionalidad requerida. A continuación se hace una breve explicación de cada una de ellas.

- **`agg_bicycle_data_by_day(dfbicycles)`**: Función que agrega los datos del uso de las bicicletas por días. La función recibe como parámetro el dataframe de la ciudad y devuelve uno nuevo con los datos resumidos. Para el resumen, la información se agrupa por estación y día. Se utiliza la función `suma` en el total de decrementos e incrementos y la función `mean` para el resto de atributos.
- **`agg_bicycle_data_by_month(dfbicycles)`**: Esta función tiene la misma funcionalidad que la anterior, pero en este caso, se agrupan los datos por estación y mes. El dataframe que se recibe como argumento es el agregado por días, ya que es de menor tamaño que el inicial y el cálculo se realiza con más velocidad. La función devuelve como resultado un data frame con la información de cada estación resumida por meses.
- **`agg_weather_data_by_day(dfweather)`**: Función que agrega los datos meteorológicos por días. La función recibe como parámetro los datos meteorológicos de la ciudad y realiza una pre-selección de los atributos necesarios

para el funcionamiento de la herramienta. Una vez seleccionados, los resume agrupándolos por día, utilizando la función `mean` para la mayoría de atributos.

- **`agg_weather_data_by_month(dfweather)`**: Función similar a la anterior, pero en este caso devuelve los datos agregados por mes. De esta forma, se pasa como argumento el dataframe agregado por días.
- **`daily_city_demand(city, dfbicycleday)`**: Función que recibe como parámetro una ciudad y el dataframe de los datos de las estaciones agregados por días. Primero se obtiene el número de estaciones que tiene la ciudad. Seguidamente, se resumen los datos por días juntando, así, la demanda de todas las estaciones. La demanda es dividida entre el número de estaciones para que pueda compararse posteriormente con otras ciudades.
- **`monthly_city_demand(city, dfbicycleday)`**: Función similar a la anterior, en este caso la demanda se agrupa por meses.
- **`subset_by_date(dataset, ini, fin)`**: Esta función recibe como parámetro un dataset y dos fechas, inicio y fin. Devuelve como resultado un dataframe con los datos reducidos según las fechas proporcionadas.
- **`transform_data(ini, fin)`**: Esta función es la que se encarga de todo el proceso de lectura de los ficheros CSV y carga de variables globales. En primer lugar, lee los ficheros y va llamando a las correspondientes funciones que hemos explicado anteriormente. La función tiene un bucle anidado para recorrer todas las ciudades y estaciones. Al final de la ejecución de la función, tenemos cargado en memoria todos los datos necesarios para el cálculo de los gráficos y la información requerida en cada uno de los apartados.

## 5.3 Sección información ciudades

---

La sección información de las ciudades está dividida en tres cajas. La primera de ellas está formada por un mapa donde se ubica la ciudad seleccionada. Junto al mapa se muestra información general de la ciudad.

La segunda, contiene información meteorológica de la ciudad en el periodo de recolección de los datos. Aquí se puede visualizar la evolución de la temperatura, el viento y las lluvias.

La tercera y última caja muestra información del uso de las bicicletas, tanto la demanda como los perfiles semanales y diarios.

A continuación se explica como se han implementado la funcionalidad de cada uno de estos elementos.

- **Apartado información general ciudades**

Para la ubicación de la ciudad en el mapa se ha utilizado la librería `leaflet`. Esta librería nos permite, a partir de la longitud y la latitud, situar un punto

en el mapa. Cuando el selector de ciudades se actualiza, el mapa se renderiza utilizando las coordenadas disponibles en el dataframe de ciudades, obteniendo como resultado una marca en el mapa.

Respecto al número de habitantes, existe un atributo en el fichero ciudades que contiene este valor de cada una de las ciudades. Este fichero lo tenemos cargado en memoria, por tanto, podemos obtener directamente el dato filtrado por el nombre de la ciudad seleccionada.

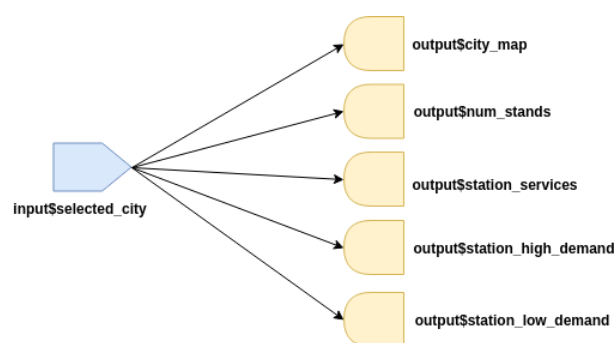
Para obtener el número de estaciones, filtramos los datos de las estaciones a partir del nombre de la ciudad seleccionada. Posteriormente, realizamos un conteo de las filas resultantes, obteniendo así el número total de estaciones.

Para el cálculo del número de viajes realizados durante el periodo de recogida de datos consideramos que un decremento es un viaje realizado por el cliente. Por tanto, sumando todos los decrementos que han habido en cada una de las estaciones, obtenemos el número total de viajes realizados en la ciudad.

La posición de la ciudad en el ranking de uso es obtenida generando un dataframe con los datos normalizados del uso de las bicicletas en ciudades. Para ello se divide la suma del total de decrementos por el número de estaciones que tiene la ciudad. De esta forma, podemos comparar la demanda de cada una de las ciudades. El dataframe resultante contiene dos columnas: ciudad y demanda media. Ordenamos los datos de manera descendente obteniendo así un ranking de uso, donde la ciudad que más utiliza la bici estará en la primera posición y la que menos en la última. Para obtener el dato correspondiente a la ciudad seleccionada, buscamos dentro del dataframe la posición que hace referencia al nombre de la ciudad. Posteriormente, mostramos el dato en la caja informativa correspondiente.

Cada una de estas cajas disponen de una función reactiva en el lado servidor. Cada vez que el selector de ciudades se actualiza, estas funciones se vuelven a ejecutar actualizando los datos de cada uno de los elementos de salida.

El siguiente diagrama muestra las relaciones existentes entre los elementos que forman este primer apartado.



**Figura 5.2:** Diagrama de relaciones entre los elementos del apartado información general de la ciudad

Como podemos observar, todos ellos tienen una conexión con el selector de ciudades del menú lateral. Un cambio de ciudad actualizará el contenido del mapa y la información asociada de interés.

#### ■ Apartado información meteorológica

La pestaña temperatura muestra un gráfico donde se puede ver la evolución de la temperatura dentro del rango temporal de nuestros datos. Para ello, hemos utilizado las librerías `ggplot2` y `plotly` junto con los datos meteorológicos recolectados.

Se muestra un gráfico de dispersión donde el eje Y representa la temperatura y el eje X el día. Cada uno de los puntos representa la temperatura media diaria. Para ello, hemos utilizado la función `ggplot` junto con las capas `geom_point` y `geom_smooth`. Pasamos como argumento el dataframe correspondiente a la ciudad seleccionada y las variables `main_temp` y `data`. Los labels de las variables y el título del gráfico con adaptados para una mejor visualización.

La segunda pestaña muestra un gráfico de dispersión donde visualizar la cantidad de precipitaciones en milímetros de agua. En este caso utilizamos también la librería `ggplot` y el dataframe con los datos meteorológico de la ciudad seleccionada. En el eje Y tenemos la cantidad de agua recogida y en el X, el día. Se ha utilizado la función `ggplot` junto con las capas `geom_point` y `geom_smooth`.

La tercera pestaña muestra la información del viento. En este caso hemos utilizado la librería `openair`. Esta librería nos ofrece una función que permite crear una rosa de vientos a partir de los datos dirección y velocidad del viento. Información que tenemos disponible en el dataset de meteorología.

Por último, mostramos las correlaciones entre la demanda y las variables lluvia, temperatura y viento. Para ello, creamos un gráfico de dispersión donde la variable Y demanda es fija, y la variable X varía según la opción que elija el usuario.

El siguiente diagrama, 5.3, muestra las relaciones existentes entre los elementos que forman este segundo apartado.

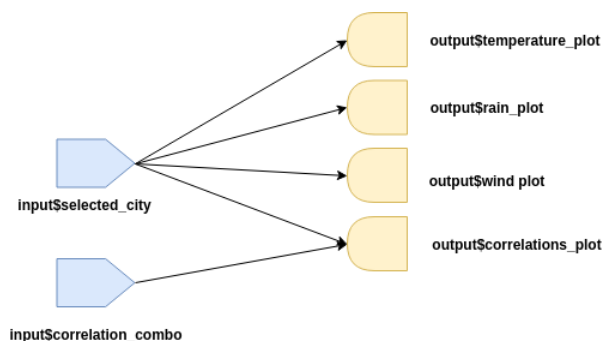


Figura 5.3: Diagrama de relaciones del apartado meteorología

#### ■ Apartado información uso bicicletas

Este apartado está dividido en dos pestañas: visualización general del uso de bicicletas y perfiles de la ciudad.

En la primera pestaña, el usuario dispone de un panel de control donde puede elegir diferentes opciones: a) si desea ver los datos de manera diaria o mensual, b) si desea ver información del tiempo y/o temperatura, y c) si desea ocultar los fines de semana. En la parte servidora se controlan las opciones que el usuario ha escogido. Creando un gráfico de barras utilizando la función `ggplot`, donde el eje de la X es el día y el eje Y la suma de decrementos normalizada. Para normalizar los datos y que sean comparables con otras ciudades se divide el total de decrementos entre el número de estaciones que tiene la ciudad. A este gráfico se añaden o eliminan capas según las opciones escogidas por el usuario. La capa de temperatura viene representada por un gráfico lineal. Para mostrar la descripción del tiempo, se diferencian los colores de las barras según la descripción del día: soleado, nublado, lluvioso.

En la segunda pestaña, se muestran los perfiles de uso de la ciudad. Para obtener un perfil de ciudad se hace la media de todos los perfiles de las estaciones que componen la red de bicicletas. Obteniendo así un perfil más general de la ciudad en conjunto. Según si el usuario selecciona el perfil semanal o diario se utilizan unos datos u otros. Estos perfiles están representados por un gráfico de barras donde el eje X muestra las horas del día (perfil diario) o las horas de la semana (perfil semanal) y el eje Y la demanda media.

El siguiente diagrama, 5.4, muestra las relaciones existentes entre los elementos que forman esta segunda caja.

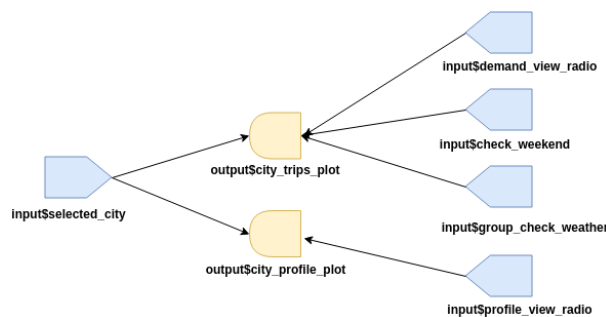


Figura 5.4: Diagrama relaciones apartado información del uso de bicicletas

## 5.4 Sección información estaciones

### ■ Apartado información general estaciones:

Este apartado está formado por un mapa de estaciones junto con información general y específica de la estación seleccionada.

El mapa se renderiza a partir de un subset de datos con la información de todas las estaciones de la ciudad seleccionada. Con estos datos y utilizando la librería `leaflet`, se ubican cada una de las estaciones. Se colorean cada uno de los marcadores según la demanda de la estación. Para ello, tenemos una función que dependiendo de la demanda asigna un color u otro al icono.



La información general hace referencia a la estación con mayor y menor demanda de todo el sistema de bicicletas de la ciudad. Para ello, se calcula la demanda media de todas las estaciones, dividiendo la suma del total de incrementos entre el número de stands. Posteriormente se ordenan de forma decreciente, donde la primera fila indica la estación con mayor demanda y la última la de menor. Con la función `head` obtenemos la primera fila y con `tail` la última.

Por otro lado, existe un `observer` que controla si alguno de los marcadores ha sido seleccionado. Obteniendo el ID cuando se el usuario realiza click, el cual hace referencia al número de la estación. Con este dato podemos obtener el número de stands y los servicios que tiene disponibles (banco y/o bonus) haciendo una consulta al fichero correspondiente.

Cuando el usuario selecciona una estación se habilita la segunda caja de esta sección.

En siguiente diagrama, 5.5, se pueden visualizar las relaciones entre los diferente elementos.

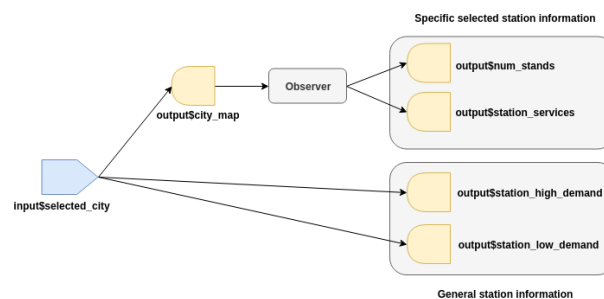


Figura 5.5: Relaciones apartado información estaciones

#### ■ Apartado información demanda estaciones:

Una vez seleccionada una estación, esta caja queda habilitada. Existen dos pestañas, la primera de ella para visualizar la demanda escogiendo un día en concreto, y la segunda para visualizar los perfiles periódicos de la estación.

En la primera de ellas, el usuario dispone de un panel de control donde puede elegir la forma de visualización de los datos: semanal o diaria, junto a un selector de fechas. Desde la parte servidora obtenemos los datos de estos elementos para crear un gráfico de barras donde mostrar la demanda de la estación. Según el modo de visualización se genera el gráfico semanal o diario. Los datos los obtenemos del dataset del uso de bicicletas, filtrado por ciudad, estación y fecha.

La segunda pestaña muestra los perfiles diarios y semanales. Al igual que en la anterior, según la opción que selecciona el usuario, se obtienen los datos y se genera un gráfico u otro.

Para la obtención de los perfiles diarios de la estación disponemos de una función que agrega los datos por estación, fecha y hora. En el proceso de agregación se utiliza la función `media`. De este modo obtenemos un dataset

de 24 filas el cual representa el perfil de la estación. Este dataset es el utilizado para la creación del gráfico en ggplot. El eje X son las 24 horas que componen un día y el eje Y la demanda.

En cuanto a los perfiles semanales, creamos una columna adicional que representa el día de la semana. Donde el día 1 es lunes y el 7 es domingo. Utilizamos esta columna junto al numero de estación y la hora para agregar los datos. Utilizando la media como función de agregación.

El dataset resultante es de 168 filas, que corresponden al número de horas que tiene una semana. A partir de estos datos se crea el gráfico correspondiente con ggplot, donde el eje de la X representa la hora semanal y el Y la demanda.

En siguiente diagrama, 5.6, se pueden visualizar las relaciones entre los diferentes elementos que conforman esta funcionalidad.

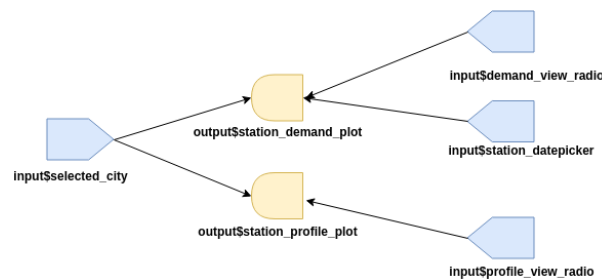


Figura 5.6: Diagrama relaciones apartado información demanda estaciones

## 5.5 Sección comparación perfiles

Esta sección nos ofrece la posibilidad de comparar perfiles tanto de estaciones como de ciudades. Siguiendo el mismo diseño que en las secciones anteriores, cada utilidad queda contenida en una caja. En este caso, la primera para comparar los perfiles de las ciudades y la segunda para las estaciones.

- Apartado comparación perfiles de ciudades:** Este apartado está formada por un panel de control donde el usuario puede seleccionar dos ciudades y el modo de visualización. A partir de esta información, se genera un gráfico lineal donde aparece la evolución de la demanda de cada una de las estaciones. Para obtener los datos se utiliza la misma función que en el apartado de visualización de perfiles de la sección información de la ciudad. Una vez tenemos ambos datasets, con los datos respectivos al perfil, se unifican en uno mismo, añadiendo como argumento el nombre de la variable.

Siguiendo esta aproximación conseguimos mostrar ambas líneas en el mismo gráfico y así podemos mostrar el nombre de cada una de las variables en la leyenda del mapa.

El siguiente diagrama, 5.7, muestra las relaciones entre los elementos de este apartado.

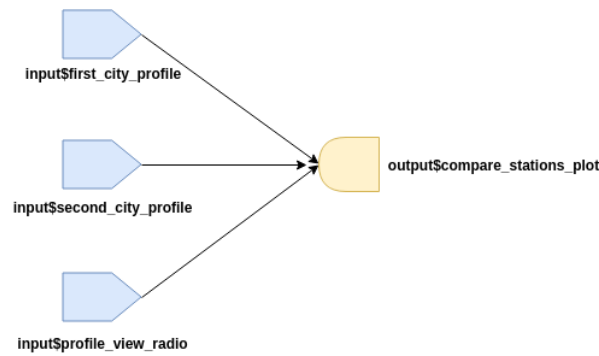


Figura 5.7: Diagrama relaciones apartado comparación perfiles de ciudades

#### ■ Caja comparación perfiles de estaciones:

En esta segunda caja se pueden comparar los perfiles de las estaciones. El usuario puede escoger ciudad y estación y compararlo con otra. Pueden ser estaciones de la misma ciudad o de otra. Al igual que en el apartado anterior, dispone de un botón radio donde elegir si visualizar el perfil semanal o diario.

Para obtener los perfiles utilizamos la misma técnica que en apartado información estaciones. Agregamos los datos por días o por semanas, según la opción elegida por el usuario y obtenemos el dataset con el perfil. Trás obtener los datos se procede a unificar ambos perfiles utilizando el nombre de variable como columna adicional. De esta forma podemos crear, al igual que en el apartado anterior, un gráfico donde visualizar ambas demandas junto con la leyenda informativa.

El siguiente diagrama muestra las relaciones entre los diferentes elementos que conforman este apartado.

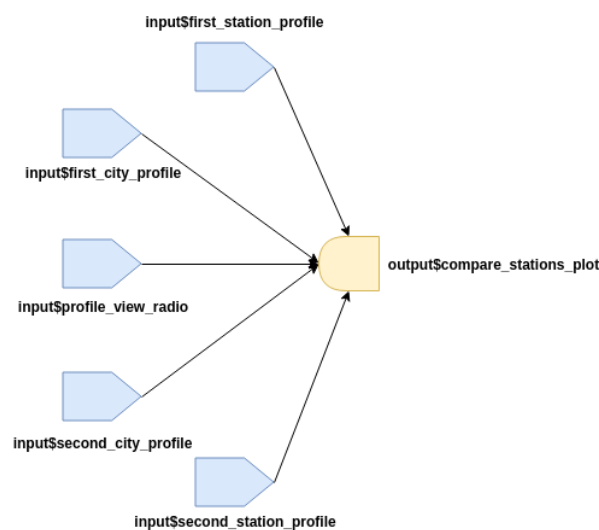


Figura 5.8: Diagrama relaciones apartado comparacion perfiles estaciones

## 5.6 Sección tiempo real

Esta sección nos permite visualizar el estado de las estaciones en tiempo real. Para ello disponemos de una función reactiva que hace llamadas GET a la API de JDCeaux cada dos minutos. La llamada se realiza utilizando el paquete jsonlite que nos permite obtener los datos JSON en formato dataframe. En esta llamada pasamos como argumento la ciudad seleccionada y la correspondiente API key que ha sido obtenida en el portal web de la empresa.

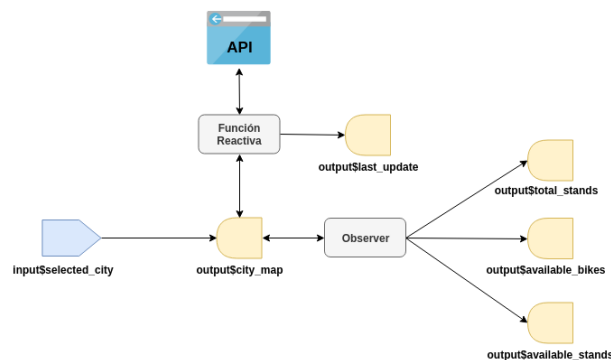
A partir de esta información renderizamos el mapa utilizando el valor de las columnas longitud y latitud. Para una mejor visualización, asignamos un color al icono de la estación dependiendo de la disponibilidad de bicicletas. Para ello, tenemos una función que recibe como parámetro el número de bicicletas disponibles y asigna un color u otro. Si hay mas de 10 bicicletas se asigna el color verde, si hay mas de 6 pero menos de 10 el color naranja y en caso contrario el color rojo.

A su vez disponemos de un observer que controla el numero de bicicletas de todas las estaciones, en el caso de que una estación tenga cero bicicletas disponibles, envía una notificación a la herramienta.

Por otro lado, disponemos de otro observer para controlar si se selecciona alguno de los marcadores del mapa. Cuando uno es seleccionado, se obtiene el número de estación y se muestra la información relativa a la estación: número de stands totales, número de bicicletas disponibles y numero de stands disponibles.

Bajo del mapa mostramos la última actualización del mapa. Para ello, disponemos de una columna en el dataframe que nos proporciona el timestamp de la última actualización. Convertimos este dato a formato fecha y lo mostramos al usuario.

El diagrama 5.9 muestra las relaciones existentes en este apartado.



**Figura 5.9:** Relaciones entre los elementos del apartado tiempo real

---

# CAPÍTULO 6

## Ejemplo de análisis de datos

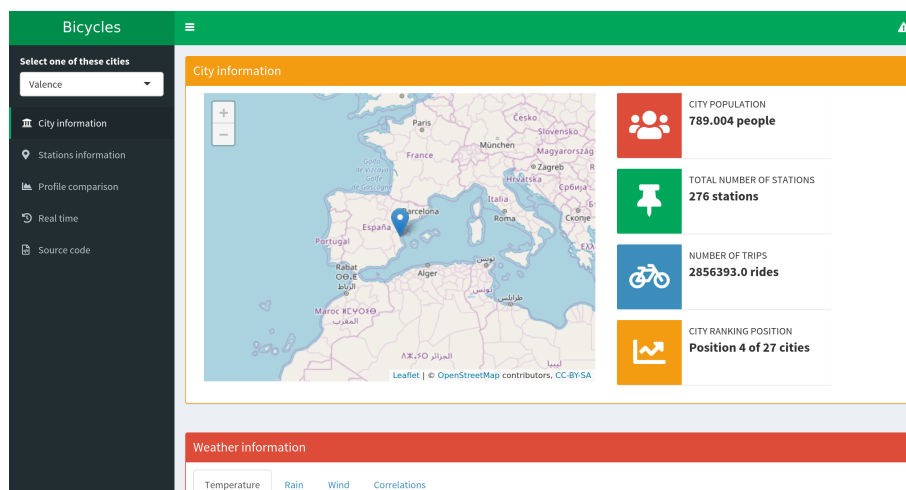
---

Con el fin de explorar las posibilidades que nos aporta la herramienta desarrollada en este proyecto, se va a realizar un análisis de los datos de la ciudad de Valencia.

Valencia es la capital de la Comunidad Valenciana, España. Está situada a las orillas del río Túria, en la costa levantina de la península ibérica, con una población total de 789.004 habitantes. Cuenta con una extensa red de bicicletas que abarca toda la ciudad, con una suma total de 276 estaciones.

Está situada en la posición 4 del ranking de uso de ciudades. Con un total de 2.856.393 viajes realizados durante el transcurso de los 7 meses de recopilación de datos.

Esta información, la podemos encontrar en el apartado información de la ciudad, tal y como se detalla en la imagen 6.1.



**Figura 6.1:** Información general de la ciudad de Valencia

Continuamos con el análisis de la ciudad de Valencia, en este caso sobre los datos meteorológicos que tenemos disponibles.

Tal y como podemos ver en la imagen 6.2, dentro del período de recopilación de datos, las temperaturas medias se han mantenido entre los 25 y los 0 grados. Siendo el período más caluroso Octubre y el más frío Enero.

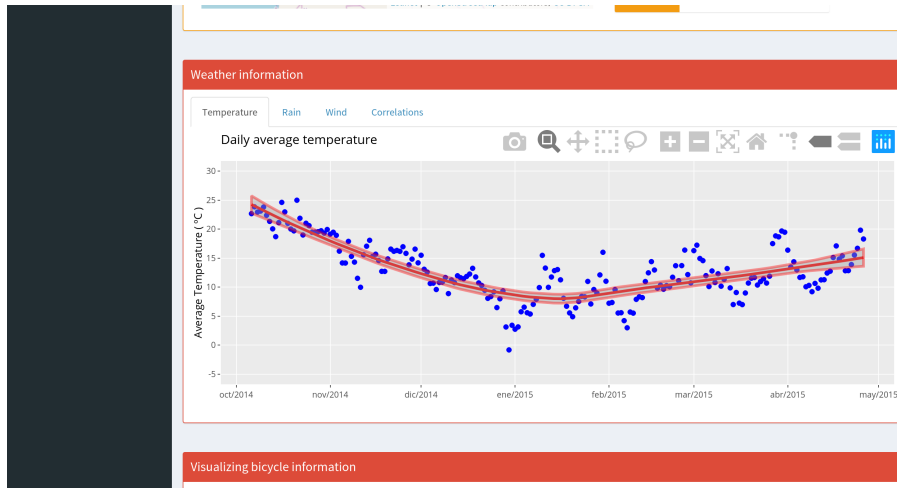


Figura 6.2: Gráfico de temperatura de la ciudad de Valencia

En cuanto a las lluvias, se rige, tal y como podemos ver 6.3 por un clima bastante seco, con lluvias de poca intensidad concentradas especialmente entre los meses de Febrero y Abril.

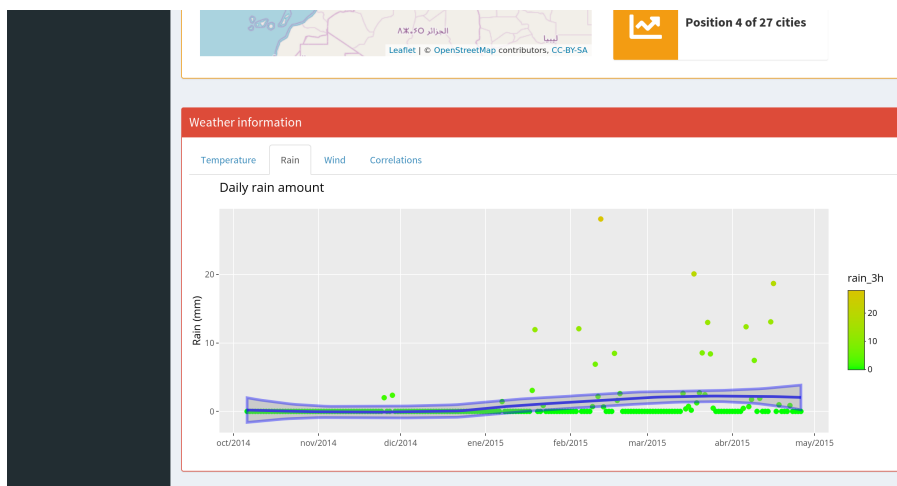


Figura 6.3: Gráfico de lluvias de la ciudad de Valencia

En cuanto al viento podemos ver en 6.4 que la mayoría de las veces va en dirección oeste, obteniendo las tasas más altas de velocidad en esta dirección. No obstante, comparando con otras ciudades, Valencia es una ciudad que no se caracteriza por fuertes rachas de viento.

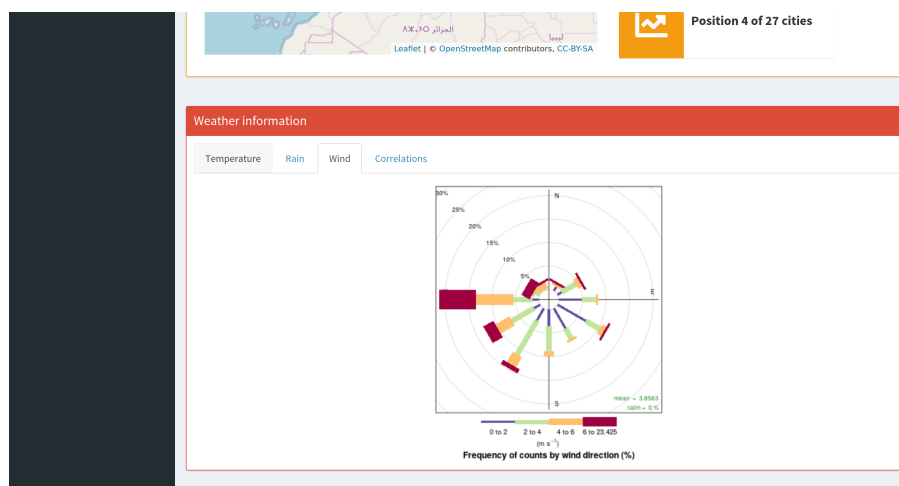


Figura 6.4: Rosa de vientos de la ciudad de Valencia

En Valencia, ni las lluvias ni el viento son factores determinantes para el uso de la bicicleta, posiblemente porque son pocos los días con lluvias o fuertes vientos. Por tanto, no existe una correlación directa entre estas dos variables y la demanda.

En cambio, tal y como podemos ver en la imagen 6.5, sí que se puede observar que existe una correlación positiva entre la demanda y la temperatura. Lo que nos indica que los ciudadanos de Valencia prefieren hacer uso de estos sistemas cuando las temperaturas son más altas o agradables.

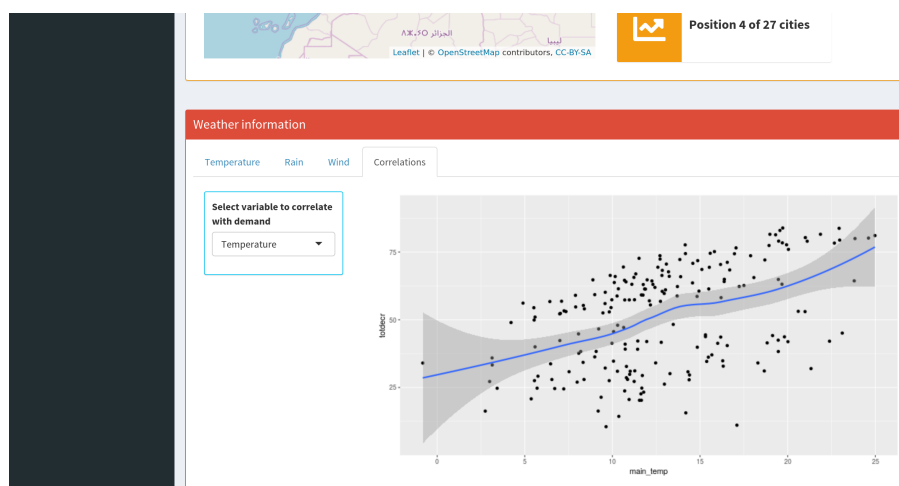


Figura 6.5: Correlación entre la temperatura y la demanda en Valencia

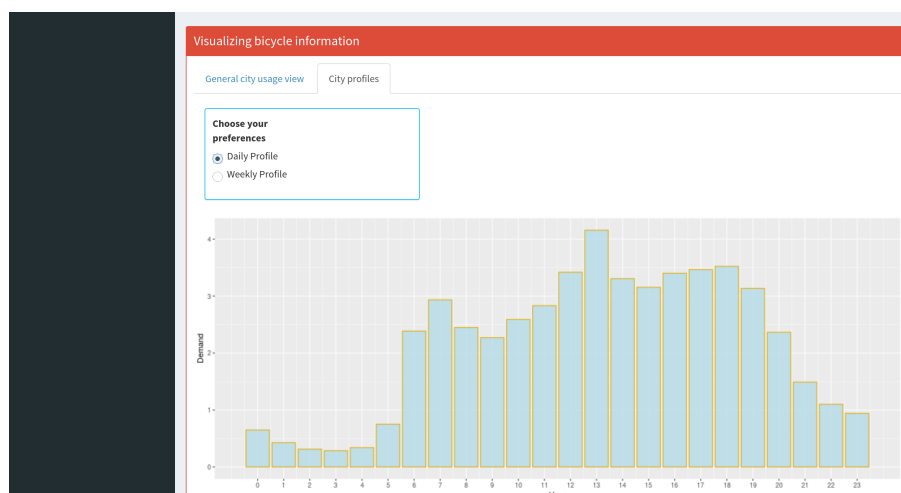
En general, Valencia es una ciudad donde predomina el sol, esto se intuye viendo las pocas lluvias recogidas y las temperaturas medias. Fiel reflejo, es el gráfico 6.6 en donde tenemos condensada toda esta información. Si analizamos el gráfico, vemos que la mayoría de días han sido soleados. Se puede observar que, efectivamente, la temperatura influye en el uso de la bicicleta. Podemos ver como alrededor del mes de Enero (el más frío de todo el año) el uso es bastante más bajo que en el resto de meses.



**Figura 6.6:** Información diaria de la demanda junto al factor meteorológico en Valencia

Si continuamos el análisis, podemos observar los perfiles periódicos de la ciudad. Este perfil recoge la media de los perfiles de cada una de la estaciones que forman la red de bicicletas.

En este caso, podemos observar en la imagen 6.7 como hay más demanda por las mañanas a las 7:00h, a mediodía, a las 13:00h y por la tarde hacia las 18:00h. Estos horarios son los que se ajustan a la mayoría de horarios laborales en España. Por lo que podemos hipotetizar que, muchos usuarios utilizan estos sistemas para ir al trabajo.



**Figura 6.7:** Perfil diario de la demanda en la ciudad de Valencia

Además, en el perfil semanal, 6.8, se puede confirmar esta hipótesis ya que, en los fines de semana, el uso de la bicicleta baja notoriamente. Es decir, se aprecia el descenso del uso de la bicicleta en los fines de semana mientras que, en los días laborales, existe una alta demanda por las mañanas, a medio día y por las tardes.



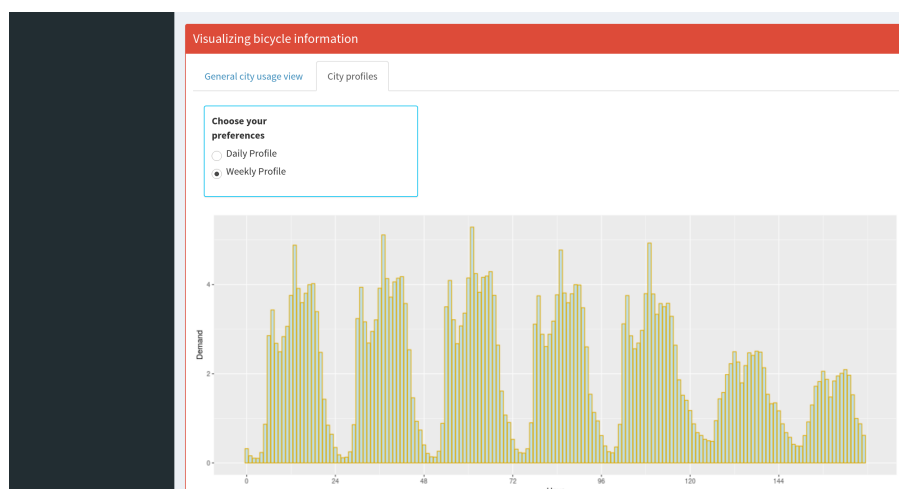


Figura 6.8: Perfil semanal de la demanda en la ciudad de Valencia

Consideremos ahora el análisis de las estaciones de Valencia. En la imagen 6.9, se puede apreciar que las estaciones más céntricas tienen mayor demanda que el extrarradio de la ciudad. Siendo la estación 96 la más utilizada y la 269 la menos utilizada. Esto nos indica que las personas utilizan estos sistemas para desplazarse por el centro, posiblemente por la dificultad de encontrar aparcamiento de vehículos, además de evitar el tráfico que se genera en las horas puntas de trabajo. Pudiendo influir, también, la concienciación por parte de la ciudadanía en conseguir una ciudad más sostenible y con menos contaminación.

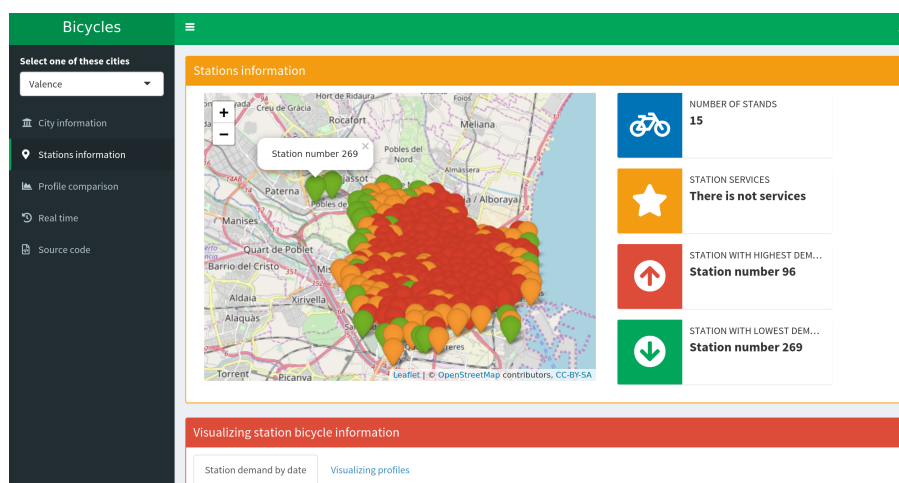


Figura 6.9: Información de las estaciones de Valencia

En esta sección podemos seleccionar una estación y analizarla en profundidad. Para ello, vamos a centrarnos en una de las estaciones de la Universitat Politècnica de València.

Como podemos ver en la imagen 6.10 esta estación está formada por 30 bicicletas y no dispone de ningún servicio de bonos ni banca. Es una estación con demanda media (color naranja). Lo que indica que, a pesar de estar dentro de la universidad, suelen haber bicicletas disponibles.

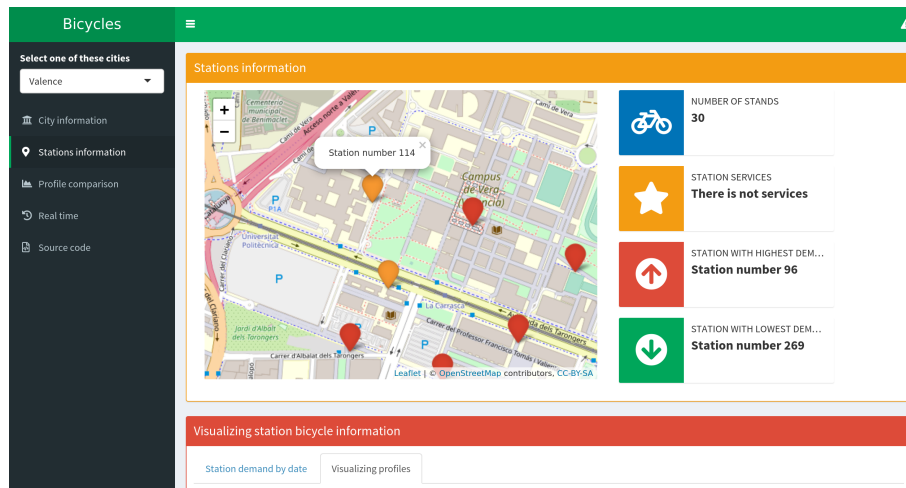


Figura 6.10: Estación dentro de la Universitat Politècnica de València

Este apartado también nos ofrece la posibilidad de visualizar la demanda en un día en concreto. En este análisis, hemos escogido el día 15 de Enero del 2015, día en el que se celebraron los exámenes en la universidad. La mayoría de viajes se concentraron a las 12:00h del medio día, 6.11, posiblemente al finalizar el primer turno de exámenes, y a las 16:00h de la tarde, en el segundo turno de exámenes.

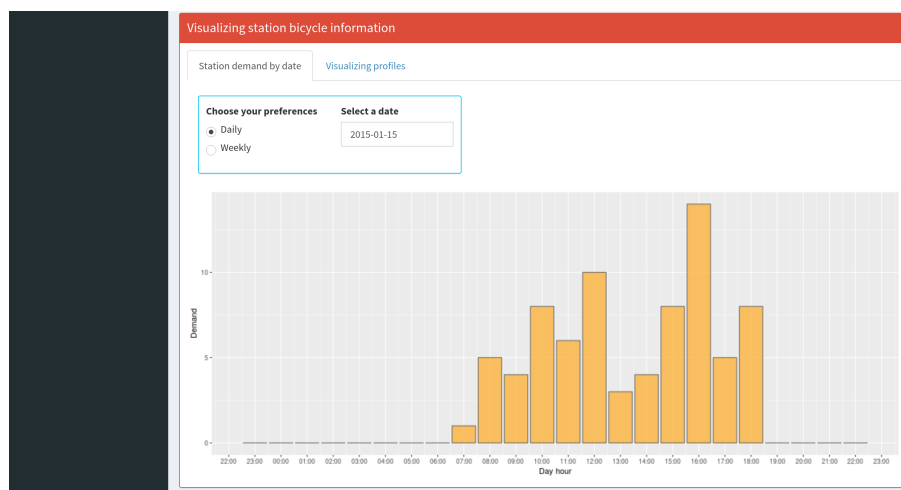
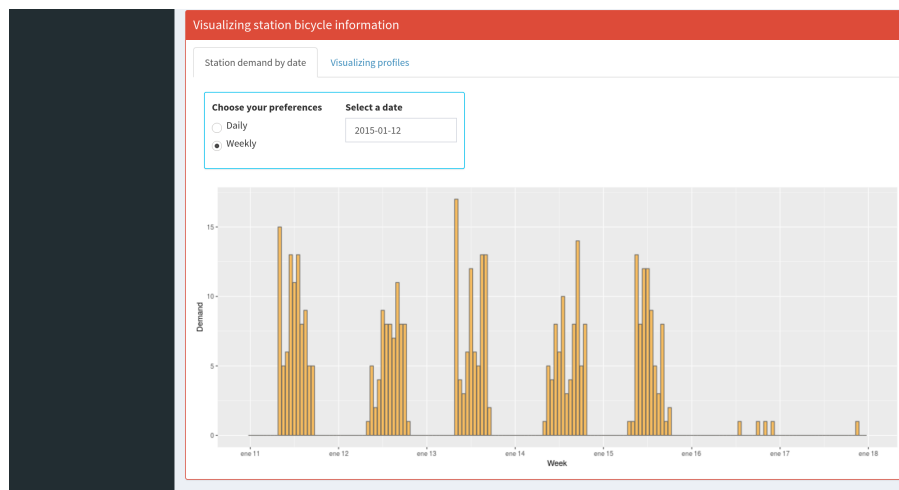


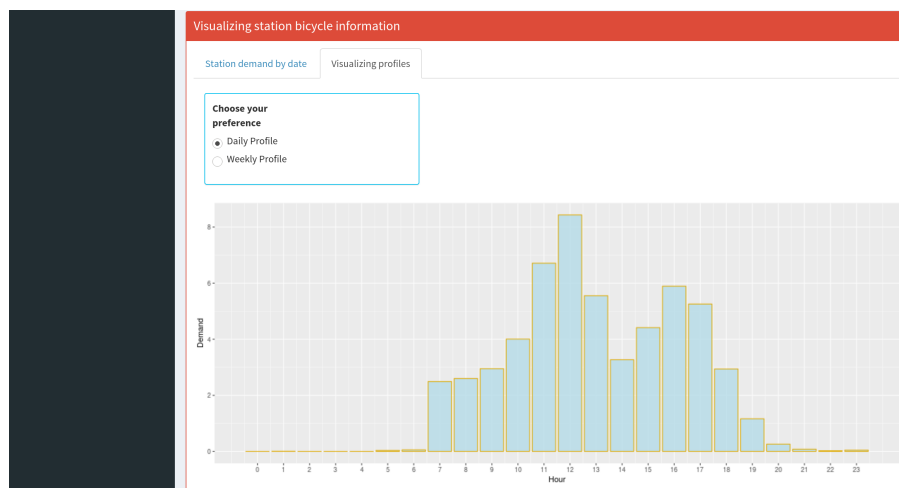
Figura 6.11: Visualización de la demanda en el día 15 de Enero del 2015

Si visualizamos la semana en conjunto 6.12, podemos ver como, en el fin de semana, prácticamente no se realizaron desplazamientos.



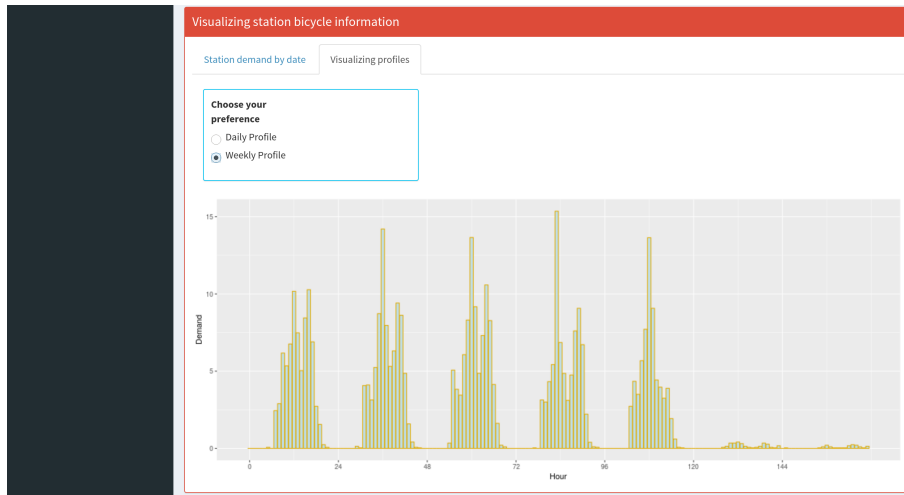
**Figura 6.12:** Visualización de la demanda de la semana del 12 de Enero del 2015

La imagen 6.12 nos puede causar curiosidad, pues existen demandas muy altas a primera hora del día. Esto no sigue la misma lógica anterior, ya que, típicamente son horas donde la gente aparca la bicicleta en la estación. No obstante, encontramos una explicación a este evento, ya que la empresa que gestiona la bicicleta, suele mover las bicicletas en estaciones que quedan completas.



**Figura 6.13:** Perfil diario de la estación de la Universitat Politècnica de València

Si visualizamos el perfil diario de esta estación podemos ver que, en general, la mayoría de viajes se realizan a las 12:00h. Es un dato bastante curioso ya que la mayoría de las clases terminan a las 14:00.



**Figura 6.14:** Perfil semanal de la estación de la Universitat Politècnica de València

Seguimos el análisis utilizando otras funcionalidades de la herramienta, para ello accedemos al apartado comparación de perfiles.

Vamos a comparar los perfiles de la ciudad de Valencia junto a otras ciudades para ver si siguen el mismo patrón de utilización.

En primer lugar, vamos a comparar el perfil diario de Valencia con una ciudad de Francia, Toulouse.

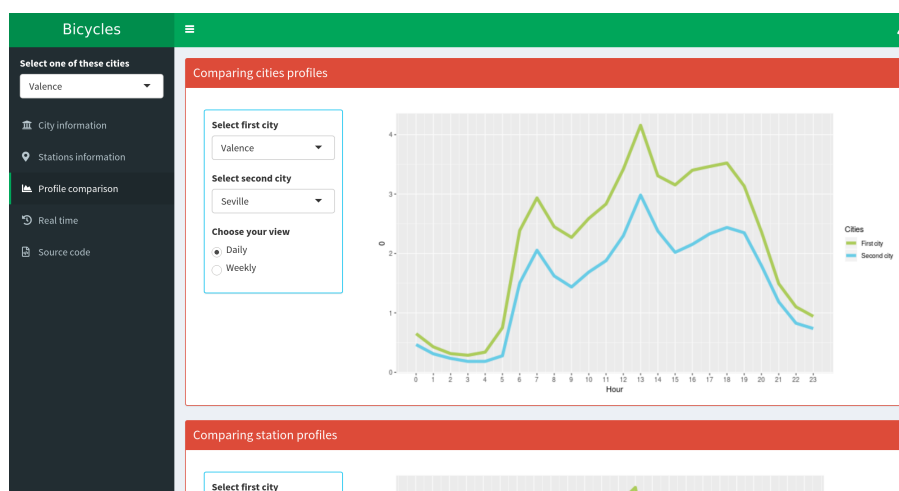
Tal y como podemos ver en la imagen 6.15 los perfiles siguen más o menos el mismo patrón. Sin embargo, se puede observar que la hora de la comida no es la misma en ambos países, en Toulouse la gente tiende a coger la bici a las 11:00h de la mañana y en Valencia a la 13:00h. Lo mismo pasa a la hora que termina el trabajo, habiendo un incremento en la demanda de 16:00h a 17:00h en Toulouse y de 17:00h a 18:00h en Valencia.



**Figura 6.15:** Comparación del perfil diario entre Valencia y Toulouse

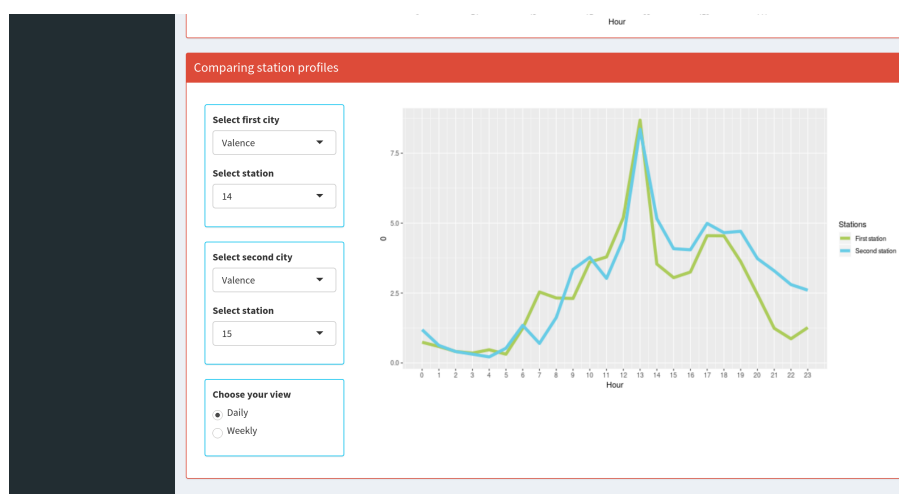
Si comparamos Valencia con otra ciudad española como, por ejemplo, Sevilla, vemos que ambas cumplen el mismo patrón. Esto puede estar reflejando el impacto de la cultura sobre el uso de las bicicletas. Pues la cultura de un país acarrea hábitos y horarios laborales importantes a tener en cuenta por parte de las empresas

que gestionan estos sistemas para garantizar una correcta adaptación y, por ende, una optimización del servicio.



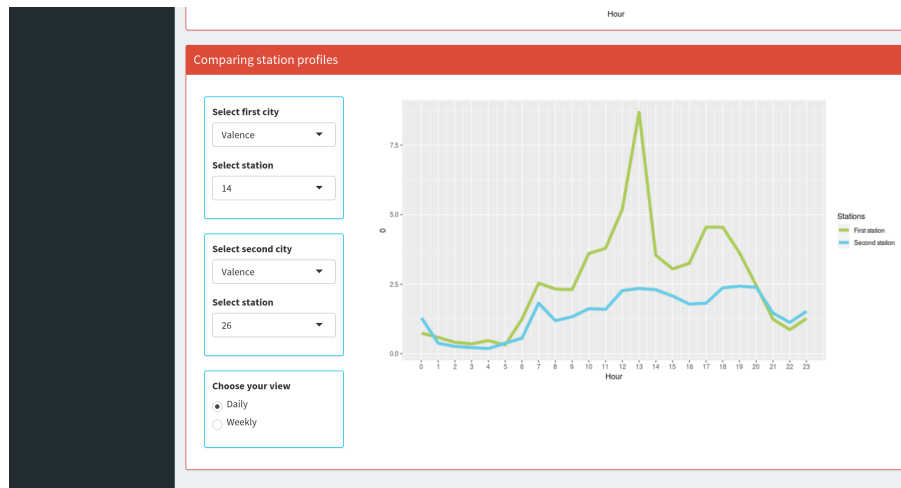
**Figura 6.16:** Comparación del perfil diario entre Valencia y Sevilla

Si nos centramos en el análisis y comparación de los perfiles de las estaciones de Valencia, podemos ver que las estaciones que están cerca tienen prácticamente el mismo perfil diario, 6.17, esto nos indica que los usuarios suelen moverse entre las estaciones cercanas. Estos resultados siguen la misma línea que los encontrados en el trabajo de Kull et al. [6].



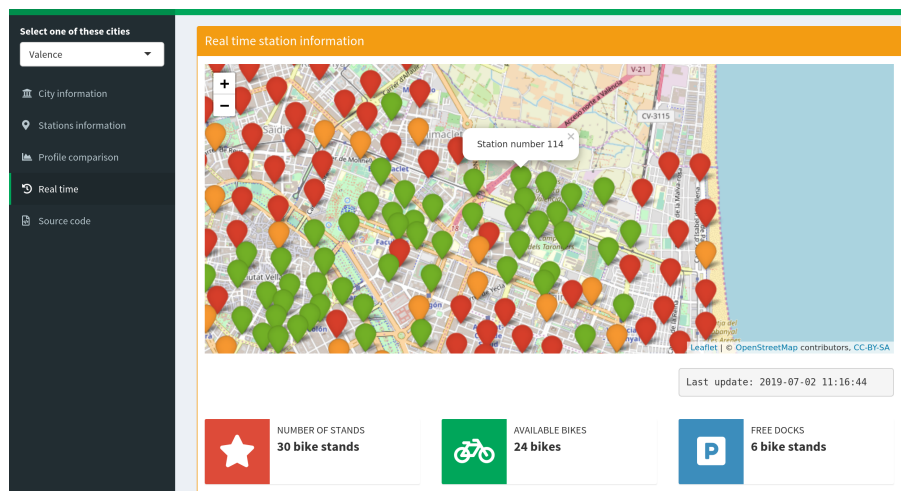
**Figura 6.17:** Comparación de perfiles entre dos estaciones cercanas de Valencia

Si comparamos estaciones que no están tan cerca, podemos ver que la demanda cambia drásticamente, este es un dato útil para las empresas de gestión, puesto que podrían realizar los balances de bicicletas teniendo en cuenta los patrones que revelan estos gráficos.



**Figura 6.18:** Comparación de perfiles entre dos estaciones distantes de Valencia

Por último, en el apartado tiempo real, podemos visualizar la demanda de cada una de las estaciones en un momento dado. En el momento de este análisis, la estación analizada de la Universitat Politècnica de València, dispone de 24 bicis disponibles y 6 stands. Este apartado, permite visualizar la demanda de las distintas estaciones y, así, planear y organizar mejor los viajes por parte de los usuarios finales. Además, recordemos que si las estaciones quedan completamente vacías el sistema nos lo notificaría.



**Figura 6.19:** Visualización en tiempo real del estado de la estación de la Universitat Politècnica de València

---

## CAPÍTULO 7

# Conclusiones y trabajo futuro

---

En este trabajo se ha desarrollado una herramienta que cubre una serie de mejoras en el ámbito del análisis de datos del uso de bicicletas en ciudades, ofreciendo una aplicación visual e interactiva que analiza con detalle los factores más influyentes. Se han abordado con éxito los objetivos planteados inicialmente, los cuales se pueden resumir en los siguientes puntos:

- Se ha estudiado el lenguaje de programación R, junto con técnicas de análisis de datos. Estos conceptos nos han permitido el desarrollo de la herramienta.
- Se han preparado los datos para poder obtener las funcionalidades deseadas en la herramienta.
- Se ha integrado información meteorológica y se ha relacionado en la demanda de las bicicletas.
- Se ha creado una sección donde se ha podido analizar en detalle cada una de las estaciones.
- Se han introducido los perfiles semanales y diarios. Pudiendo compararlos entre ciudades y estaciones. Visualizando, así, las diferencias existentes entre las diferentes ciudades.
- Hemos mantenido la utilidad de visualizar la demanda en tiempo real, añadiendo nuevas funcionalidades como la distinción por colores de la demanda de las estaciones en el mapa.
- Finalmente, se ha realizado un ejemplo de uso donde se puede ver la utilidad de la herramienta. De forma visual, se puede analizar rápida e intuitivamente cómo afecta el momento del día y el factor meteorológico en la demanda, así como las influencias de la cultura en las franjas horarias de utilización de las bicicletas.

Esta herramienta ofrece un marco de trabajo que puede ser ampliado con multitud de posibilidades para la representación de este tipo de datos. Aunque no se han abordado por quedar fuera del alcance del proyecto, a continuación se proponen algunas sugerencias para futuro trabajo:

- **Incorporación de nuevos datos:** La herramienta cuenta con datos de estaciones dentro de 27 de ciudades. Sería interesante la recopilación de más datos, por ejemplo, de más ciudades, para la ampliación de la herramienta.
- **Distintos tipos de sistemas de movilidad:** La herramienta se centra en el análisis del uso de bicicletas. No obstante, recientemente está habiendo un crecimiento en otros sistemas de movilidad, como pueden ser las motos eléctricas, por ejemplo, Moving en Valencia. Una posible ampliación sería la integración de alguno de estos sistemas, ofreciendo así una herramienta general e integrada de visualización de sistemas de movilidad sostenibles.
- **Predicción:** Para tener una aplicación completa, sería interesante añadir la utilidad de poder predecir la demanda en el futuro. Para ello, se podrían implementar técnicas de inteligencia artificial donde utilizar los datos del pasado, junto a los del tiempo real, para la obtención de predicciones en el futuro.



# Bibliografía

---

- [1] Nam, T., and Pardo, T. A. Conceptualizing smart city with dimensions of technology, people, and institutions. In *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times* ACM (pp.282-291). (2011, June).
- [2] DeMaio, Paul. Bike-sharing: History, Impacts, Models of Provision, and Future. *Journal of Public Transportation*, 12 (4): 41-56. 2009
- [3] Schlote, A., Chen, B., Sinn, M., and Shorten, R. The effect of feedback in the assignment problem in shared bicycle systems. In *International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 960–961, Dec 2013
- [4] Frade, Inês and Ribeiro, Anabela. Bicycle sharing systems demand. *Procedia - Social and Behavioral Sciences*, 111:518–527, 2014
- [5] Fanaee-T, Hadi and Gama, Joao. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pp. 1–15, 2013.
- [6] Kull, Meelis, Ferri, César, and Martínez-Usó, Adolfo. The bike-sharing demand, 2015. URL: <http://dmip.webs.upv.es/BikeSharingDemand/>
- [7] Davis, L. S. Rolling along the last mile: Bike-sharing programs blossom nationwide. *Planning*, 80(5), 10-16. 2014.
- [8] Fishman, Elliot. Bikeshare: A review of recent literature. *Transport Reviews*, pp. 1–22, 2015.
- [9] K-W Moon. *Learn ggplot2 Using Shiny App, Use R!*. Springer International Publishing AG 2016.
- [10] Dix, A. *Introduction to information visualisation*. In *Information Retrieval Meets Information Visualization* (pp. 1-27). Springer Berlin Heidelberg <http://link.springer.com/book/10.1007/978-3-642-36415-0/page/1>
- [11] Bike-Sharing systems categorisation. Consultado en [https://en.wikipedia.org/wiki/Bicycle-sharing\\_system#Categorisation](https://en.wikipedia.org/wiki/Bicycle-sharing_system#Categorisation). [Último acceso: Junio 2019]
- [12] O'Brien: Bike share map for across the World. Consultado en <http://bikes.oobrien.com/#zoom=3&lon=60&lat=40>. [Último acceso: Mayo 2019]

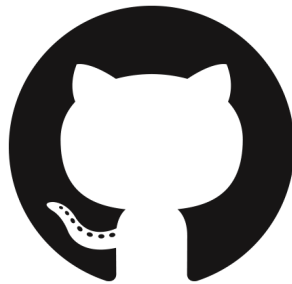
- 
- [13] Citybikes: Bike sharing data for everyone. Consultado en <https://citybikes.es/>. [Último acceso: Mayo 2019]
  - [14] Bike Visualization. The most complete bike sharing visualization and analysis website. Consultado en <https://www.visualization.bike/>. [Último acceso: Junio 2019]
  - [15] Shiny: Get started. Consultado en <https://shiny.rstudio.com/tutorial/>. [Último acceso: Junio 2019]
  - [16] Split app code across multiple files. Consultado en <https://github.com/daattali/advanced-shiny/tree/master/split-code>. [Último acceso: Junio 2019]
  - [17] Introduction to R. Consultado en <https://www.r-project.org/about.html>. [Último acceso: Junio 2019]
  - [18] Stackoverflow developer survey results. Consultado en <https://insights.stackoverflow.com/survey/2019>. [Último acceso: Junio 2019]
  - [19] JCDcaux developer. Consultado en <https://developer.jcdecaux.com/#/opendata/vls?page=dynamic> [Último acceso: Abril 2019]
  - [20] OpenWeatherMap API. Consultado en <https://openweathermap.org/api> [Último acceso: Abril 2019]
  - [21] Paquete R ggplot2. Consultado en <https://en.wikipedia.org/wiki/Ggplot2> [Último acceso: Junio 2019]
  - [22] Infograma comparativo R y Python Datacamp. Consultado en <https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis> [Último acceso: Junio 2019]
  - [23] Leaflet maps Consultado en <https://leafletjs.com/> [Último acceso: Junio 2019]

---

## APÉNDICE A

# Obtención del código fuente de la herramienta

---



El código fuente de la herramienta, junto con los datos meteorológicos y del uso de las bicicletas, están disponible en un repositorio público en Github.

Tenemos dos opciones para obtener todos los ficheros, o descargarlos en formato zip o clonar el repositorio haciendo uso del comando **git clone**.

Para la obtención del zip entramos al siguiente enlace y pulsamos el botón **Download ZIP**.

<https://github.com/vicennt/bicycles-use-analysis>

Si preferimos obtener el repositorio usando la terminal Linux, nos situamos en el directorio deseado y ejecutamos la siguiente instrucción:

```
$ git clone https://github.com/vicennt/bicycles-use-analysis.git  
bicycletool
```

Este comando creará un nuevo directorio llamado **bicycletool** donde tendremos el código y los datos disponibles.

Una vez tenemos el código en local, podremos acceder al código fuente de la herramienta.

Para ejecutar la aplicación tenemos que abrimos RStudio y ejecutamos el fichero `app.R`.