

HERRAMIENTA PARA LA VISUALIZACIÓN DE LA MOVILIDAD URBANA DE VILANOVA I LA GELTRÚ

Andrés Prieto González

Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú

Resumen

Este proyecto tiene como objetivo el desarrollo de una herramienta para modelar y visualizar la movilidad urbana de una ciudad, en este caso Vilanova i la Geltrú.

Para ello, desarrollaremos un diseño organizado en dos bloques: uno de generación de rutas de los ciudadanos y otro para la visualización de estas. Gracias a esta estrategia, podremos desacoplar el esfuerzo computacional haciendo que la visualización requiera un mínimo trabajo y sea más fluida.

Esta herramienta, por lo tanto, nos permitirá generar un modelo de movilidad para cualquier ciudad y poder visualizarla en un espacio de tiempo determinado, siempre y cuando dispongamos de la información requerida.

Palabras clave- *Smartcities*, movilidad urbana, vilanova i la Geltrú, visualización, simulación

1. Introducción

Las *smartcities* son un concepto emergente, el cual consiste en incorporar tecnología en las ciudades para ofrecer cualquier tipo de servicio destinado o no a sus habitantes, como puede ser una administración inteligente del agua de la lluvia para el riego de zonas verdes. El objetivo, por tanto, de las *smartcities*, es la mejora del estado actual de la ciudad en cualquier aspecto que se nos pueda venir a la mente. En una *smartcity*, si queremos ofrecer un buen servicio al ciudadano, se requiere disponer de abundante capacidad de cálculo ya que habrá multitud de ellos que demanden cualquier tipo de servicio que ofrezcamos.

Fog computing es una tecnología aún en desarrollo que mejora el paradigma de cloud, ofreciendo lo que a *cloud computing* le falta, rapidez. La idea detrás de *fog computing* es que el cálculo se realiza lo más cerca del ciudadano (*edge*), sin necesidad de tener que subir hasta el servidor centralizado (*cloud*), lo que llevaría mucho tiempo, perdiendo así la noción de instantaneidad.

Entre los diferentes problemas que presenta la tecnología de fog computing se encuentra la de conocer la cantidad de dispositivos que puede haber en una ciudad real, y, por

tanto, la eventual capacidad de cálculo de los dispositivos capaces de realizar procesamiento (*fog*).

El objetivo de este proyecto es el diseño y la implementación de una herramienta gráfica con la que podamos ver el flujo de personas y, por lo tanto, la concentración de dispositivos *fog* (*smartphones*, *on-board computers*) en el transcurso de un día completo en una ciudad, aplicada en nuestro caso Vilanova i la Geltrú, una ciudad de la provincia de Barcelona. Aunque en la práctica esta herramienta, podría ser extrapolable a cualquier ciudad.

Esta visualización, por tanto, mostrará una estimación del comportamiento de la totalidad de sus habitantes durante un día laborable cualquiera, los cuales realizarán una ruta que corresponderá al desplazamiento de un día de cada persona. La herramienta nos ofrecerá una idea clara del estado en cada momento de la ciudad, lo que, nos permitirá entender como Vilanova i la Geltrú se mueve y, la posibilidad de localizar potenciales concentraciones *fog* para un eventual procesamiento e incluso, zonas de congestión de tráfico.

La simulación se realizará a partir de información pública y, en caso de no disponer de todo lo necesario, se estimará la que nos falte aproximándose lo máximo a la realidad. Además, la idea no es solo realizar únicamente la simulación para Vilanova i la Geltrú, sino crear un modelo con el que podamos simular cualquier ciudad teniendo una cierta cantidad de información real o teórica. Debo añadir que, aunque anteriormente he dicho que la información es de dominio público, ésta podría provenir de teléfonos inteligentes, tabletas, coches inteligentes y, en definitiva, cualquier dispositivo con conectividad que disponga de localización GPS.

2. Especificaciones

En este apartado se analizarán toda la serie de especificaciones que queramos que tenga nuestro trabajo, guiadas por el estudio de los diseños similares que se han realizado en este ámbito. Una vez tengamos claro que es lo que queremos de nuestro proyecto, podremos entender y justificar cuales de todas las especificaciones serán posibles y cuáles no.

Así que, en nuestro caso, estas serán las especificaciones con las que nos hemos encontrado:

- Simular el movimiento de toda la población de Vilanova i la Geltrú de manera realista
- Visualizar mediante un mapa de calor para poder entender las concentraciones de gente
- Utilizar toda la información real pública disponible sobre el comportamiento de los ciudadanos, y estimar la información que nos falta con la máxima precisión posible.
- Establecer una simulación cronológica que represente el paso del tiempo
- Conseguir que la visualización se vea fluida
- Permitir al usuario ajustar la visualización permitiéndole avanzar o retrasar la simulación en el tiempo
- Permitir al usuario modificar la velocidad de la visualización, pudiéndola hacer más rápida o más lenta según convenga
- Realizar una exhausta documentación sobre el proyecto para su futuro desarrollo y mejora
- Desarrollar un proyecto de código abierto, es decir, accesible por todo el mundo de forma gratuita.

Comentar que, todas estas especificaciones han sido cumplidas, pero, en el caso de las opciones de poder permitir al usuario modificar la visualización no han podido ser desarrolladas y por tanto se moverán al apartado de trabajo futuro. Aclarar que, aunque no hayan sido cumplidas, la forma en que se ha implementado el proyecto no serán un gran problema de cara al futuro desarrollo y mejora del proyecto.

3. Objetivos

El objetivo global de este proyecto es claro y conciso, consiste en el desarrollo de una herramienta gráfica para la observación de la movilidad urbana durante el transcurso de un día. Por tanto, gracias a la visualización de estas transiciones de toda la masa de población podremos llegar a conclusiones que resultarán útiles para entender cómo funciona una ciudad (por ejemplo, ciertas horas de concentración en el centro, mucho movimiento en las afueras cuando amanece).

Este objetivo global se puede estructurar en varios objetivos parciales, los cuales se pueden dividir de la siguiente forma:

- Proponer un modelo básico para la simulación del flujo de personas en ciudades a gran escala.
- Diseñar una estructura que pueda soportar todo el cómputo de la simulación y la visualización.

- Mostrar visualmente las zonas de concentración de personas en la ciudad.
- Comprobar que simular una ciudad es posible a partir de escasa información y aun así poder conseguir resultados significativos.
- Demostrar que teniendo toda la información de una ciudad podríamos ofrecer multitud de servicios como zonas de descongestión de tráfico según la zona y la hora gracias a la similitud con la situación real.
- Ofrecer una base para el estudio de la tecnología *fog computing* aplicada a una ciudad inteligente gracias a un mejor conocimiento la situación de los dispositivos en la ciudad.

4. Estado del arte

En este apartado se analizará el estado actual de los proyectos relacionados con las visualizaciones y simulaciones de flujos urbanos. Algunos de estos proyectos, están enfocados a un enfoque empresarial mientras que otros, tienen un fin educativo. Como es obvio, los que tienen un enfoque empresarial, como puede ser *Via* tienen unos precios que nadie menos empresas se podrían permitir.

Por esta razón existen otras alternativas que son enfocadas a fines educativos, como puede ser *SUMO (Simulation of Urban Mobility)*, pero, ninguna de estas alternativas está enfocada a simulaciones a gran escala, por esta razón se encontró la necesidad de la creación de un nuevo tipo de simulador, cubriendo así esta necesidad.

A continuación, se puede observar una tabla comparativa con todas las opciones que se han considerado a modo de esquema para entender que nos ofrecían cada una de ellas:

Proyecto	Capacidad de agentes	Mapa congestión	Sentido de la hora	Código abierto
Unity3D	-	✓	✗	✗
Via	✓	✓	✓	✗
SUMO	✗	✗	✓	✓
NYC Taxi	✗	✗	✓	✓

Tabla 1. Ilustración de las alternativas existentes

En consecuencia, nuestro proyecto final será una mezcla de los factores positivos mencionados en la Tabla 1, obteniendo así el mejor resultado al realizar la combinación de los proyectos existentes.

5. Diseño

En este apartado se explicará la arquitectura de nuestro proyecto y las ideas que nos han llevado hasta él. Este apartado tiene un objetivo claro y conciso y es dar a entender la estructura con una visión general pasando por todos los módulos que se implementen. Esta parte es

extremadamente importante ya que un buen diseño de un proyecto conlleva un buen resultado.

El diseño del proyecto se ha realizado para que sean bloques separados pero conectados mediante una base de datos, así poder distribuir la computación necesaria en términos de coste temporal y de eficiencia. Al diseñarlo de esta forma, podemos asegurar que la visualización tendrá las siguientes características:

- Visualización simplificada
- Visualización con poco coste computacional
- No será una visualización en tiempo real ya que la generación conlleva un tiempo computacional
- Una vez finalizada la etapa de generación, podremos visualizarla tantas veces como deseemos

En la siguiente figura, podemos observar una ilustración de la idea de este diseño.

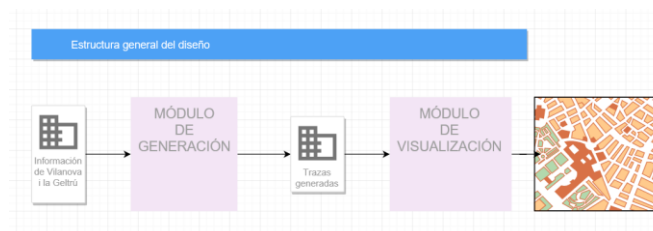


Fig. 1. Diseño de nuestro proyecto

Al ver esta estructura, podremos observar que está dividido en 3 partes, información de Vilanova i la Geltrú, un módulo de generación y un módulo de visualización:

- **Información de Vilanova i la Geltrú.** Esta parte es donde especificaremos la información de Vilanova i la Geltrú que permitirá crear un modelo de comportamiento y movilidad para los ciudadanos. Es por tanto que, contra mejor calidad de información tengamos, mayor realidad lograremos observar en la visualización. Por esta razón, es de vital importancia que en el futuro a esta parte se le dedique un hilo de investigación por sobre de todas las otras partes, ya que aquí reside la realidad de nuestras visualizaciones. He de aclarar que esta parte, en caso de cambiar de ciudad, sería la parte que deberíamos modificar.
- **Módulo de generación.** Será el encargado de realizar, para cada ciudadano y a partir de la estimación de su movilidad, la generación de las trazas para que, a posterior, el módulo de visualización las muestre de forma gráfica.
- **Módulo de visualización.** Tendrá la función de visualizar las trazas que se hayan obtenido en el módulo de generación.

Respecto a la información de Vilanova i la Geltrú hará falta entender que es lo que necesitaremos, para que lo necesitaremos y como lo utilizaremos. Para entender el funcionamiento de una ciudad primero hace falta entender a su población, por esta razón clasificaremos a la población en función de su comportamiento esperado. La clasificación

utilizada en este proyecto ha estructurado en base a las edades de sus habitantes ya que ésta marcará sus rutinas:

- **Edad escolar.** Cada persona que pertenezca a esta edad, conoceremos su rutina en gran parte ya que, la rutina de alguien en edad escolar vendrá dada por, ir al colegio y, muy posiblemente, ir a una zona de ocio a posterior.
- **Edad de trabajar.** Como en el caso de la edad escolar, la persona irá a trabajar y, a posterior ir a una zona de ocio para relajarse después de haber ido a trabajar (estos parámetros se ajustarán más o menos a la realidad en función de la información, igual que en el caso de los que están en edad escolar). Pero, aquí falta aclarar un aspecto, y es el paro. En nuestro caso, debido a que no conocemos la rutina de esta gente, se interpretará como que tiene la misma rutina que alguien que se encuentra en edad de jubilación.
- **Edad de jubilación.** En este caso se interpretará que su actividad diaria es una consecución de zonas de ocio.

Y, la información que necesitaremos será la siguiente:

INFORMACIÓN	NECESIDAD
Barrio	Número de habitantes, densidad de población, coordenadas que lo limitan
Centro educativo	Horario, coordenadas donde se encuentra, edades que lo limitan (por ejemplo, 3-16, incluye parvulario, educación primaria y secundaria)
Centro de trabajo	Horario, coordenadas donde se encuentra
Zona de ocio	Coordenadas donde se encuentran las zonas de ocio más populares

Tabla 2. Información necesaria de Vilanova i la Geltrú

En caso del módulo de generación, este no será un único módulo, sino que será una consecución de partes que tendrán la función de la generación de trazas y su preparación para la posterior visualización. Este módulo de generación tiene la siguiente estructura:



Fig. 2. Diseño del módulo de generación

Cada una de las partes tendrá las siguientes funciones:

- **Generación de rutas GPS y horarios.** Esta parte será la encargada de unificar y hacer uso de la información que se ha especificado anteriormente. Este, dará forma a la información generando así todas las rutas y horarios para todos los habitantes. Estos, se escribirán en dos trazas diferentes.
- **Traducción de coordenadas GPS a coordenadas del marco de simulación.** En este paso, lo que se hará es la creación del marco de simulación. En pocas palabras, este marco será donde proyectemos toda la visualización (por encima del mapa, es decir, no es el mapa). Además de esta traducción, se añadirán puntos intermedios en las rutas para crear una sensación de fluidez en la ruta realizada (para que no existan saltos en la ruta).
- **Registro de ejecución de las rutas respetando los horarios.** El objetivo principal de esto será la ejecución de la visualización, pero en este caso, no lo visualizaremos, sino que lo registraremos en un fichero. Hace falta aclarar que, la ejecución de esta será respetando los horarios especificados previamente, por tanto, el registro será la ejecución como si fuera la visualización. Pero, gracias a que es registrado en un fichero, podemos conseguir simpleza en la visualización (el módulo de visualización simplemente realizará la lectura de este) y, podremos tratar la traza para conseguir en este caso el mapa de calor, el cual realizaremos en el siguiente paso.
Hace falta aclarar que, con el registro obtenido se podrán obtener las informaciones que deseemos. Por ejemplo, si queremos analizar el tráfico de una calle concreta, simplemente buscaremos en cada línea de ejecución (que equivale a un momento del día) cuantas personas se encuentran en ésta, consiguiendo así el tráfico de la calle. Por tanto, será en este registro donde tendremos que ir cuando queramos analizar de manera matemática el estado real de la ciudad, no en la visualización, donde simplemente lo veremos de forma gráfica.
- **Generación del mapa de calor dinámico.** En este paso lo que se realizará es simplemente la obtención del registro total y, analizar por cada línea la concentración de cada zona. Por lo que, en función de la congestión de cada zona, se le escribirá a cada uno de los agentes el color que le corresponde en cada momento del día.
- **Generación del cálculo del tamaño de cada línea de la traza final.** Éste, será el último paso de nuestro módulo de generación. El cálculo que realizaremos en este caso nos causará una increíble optimización en la visualización, ya que, lo que realizaremos con este cálculo es decirle al módulo la cantidad de información que se tiene que cargar en cada momento. Gracias a esto, lo que conseguiremos es optimizar el uso de memoria de una manera muy considerable, haciendo que la visualización sea ligera al no tener que cargarse toda la información.

El módulo de visualización tendrá la función de realizar la lectura de las trazas calculadas en el módulo de generación

y de su visualización sobre un mapa real de la ciudad. Como se ha comentado anteriormente este bloque se diseñó para ser simple y con poco coste computacional, por ello solo consta de un simple paso, el cual consistirá en la lectura de las trazas generadas en el bloque de generación. La lectura del fichero con el tamaño de todas las líneas consistirá en ir cargando cada línea y visualizarla (guiándonos del fichero de tamaños calculado en el último paso del módulo de generación), de forma que en memoria solo tendremos cargado el mapa de calor, el fichero de tamaño de las líneas y la línea que estamos visualizando. Por esta razón, simular grandes ciudades no será problema.

6. Implementación

La implementación de este proyecto se ha realizado siguiendo el diseño previamente explicado.

La idea del módulo de generación es que, a partir de la entrada de información de la ciudad, saque las trazas necesarias para su visualización. Pero es necesario entender como se ha implementado cada una de sus partes:

- **Generación de rutas GPS y horarios.** Se ha utilizado *OSRM (Open Source Routing Machine)* el cual generará las rutas a partir de los puntos proporcionados. Estos puntos vendrán dados por la rutina de la persona, por ejemplo, para un estudiante será la siguiente:
 - La persona sale de casa para entrar a su centro escolar
 - Después de realizar las clases, acude a su casa a comer
 - Una vez ha acabado de comer, se dirige a una zona de ocio
 - Después de una estancia en la zona de ocio, se dirige a su casa para finalizar su día

Esta rutina, se traducirá en una serie de coordenadas, las cuales, se pasarán al servidor *OSRM* (petición mediante *cURL*) y nos retornará en forma de *JSON* la sucesión de coordenadas que formarán la ruta. Además, los horarios también se marcarán en función de su rutina. Por ejemplo, siguiendo el caso del estudiante:

- Sale de casa para llegar puntual a la entrada del colegio (si entra a las 8:30AM y tiene 5 minutos de camino, saldrá a las 8:25AM).
- Se quedará en el colegio hasta la hora de su salida, por ejemplo, hasta las 14:30 PM
- Cuando llegue a su casa, estará un tiempo aleatorio entre un rango de entre 1 hora y 2 horas.
- La estancia, igual que en el caso anterior, será aleatorio entre 1 y 2 horas.

- **Traducción de coordenadas GPS a coordenadas del marco de simulación.** Para la implementación de esta parte, previamente hace falta entender que lo haremos encima de un marco de simulación (*canvas*). Esta parte, se ha implementado en *Javascript* a partir de las librerías *D3.js* y *StreamSaver.js*. A partir de *Javascript* realizaremos las traducciones de coordenadas (pasamos de coordenadas GPS a coordenadas del *canvas*) y, a posterior, realizaremos la generación de puntos intermedios en las rutas a partir de la librería *D3.js*. Estos puntos intermedios, como se ha explicado previamente, darán fluidez a la ruta, ya que los agentes no realizarán saltos dentro de la ruta. Para entender esta idea, por ejemplo, veamos la Figura 3 para observar una ruta generada (sin puntos intermedios).

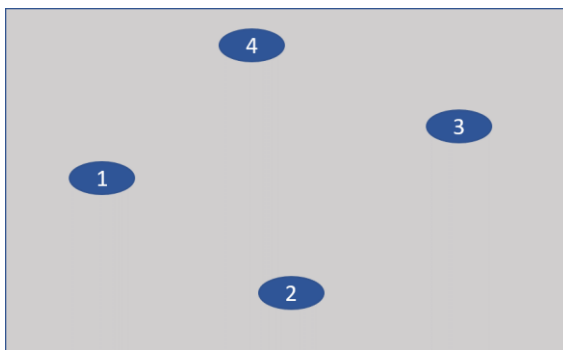


Fig. 3. Canvas con una ruta reflejada

Y, como se puede entender, si nuestro agente visualmente realiza saltos, perderá toda su continuidad. De esta forma, los puntos intermedios tendrán que ser añadidos. Estos, los podemos ver en color verde en la Figura 5.

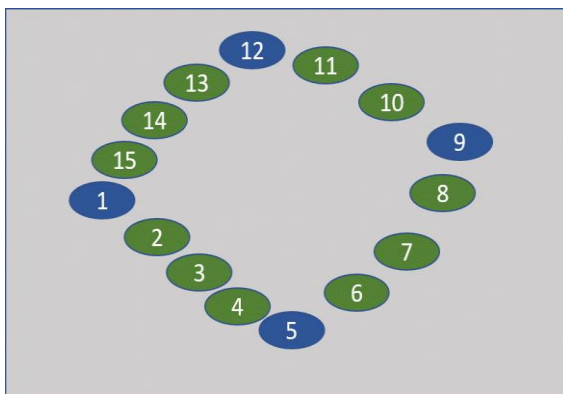


Fig. 4. Canvas con una ruta reflejada y puntos intermedios

Y, una vez tengamos generado todo, lo descargaremos mediante la librería *StreamSaver.js*.

- **Registro de ejecución de las rutas respetando los horarios.** Este registro tendrá múltiples utilidades. Entre ellas se encuentran las siguientes:

- Facilitar la vida a la visualización.
- Obtención de los datos que visualizaremos en un fichero. Gracias a esto, podremos analizar de forma matemática el estado de la ciudad en cada momento.
- Posibilidad de crear extensiones a nuestro proyecto. En nuestro caso, hemos aplicado el mapa de calor dinámico. Esto, se podría extender a gráficas que analizan el estado de cada barrio, tráfico en cada calle, etc.

Este paso será el que se encargará de generar el fichero final para la visualización. Lo que se realizará en este caso será hacer la ejecución de la simulación, pero en vez de visualizarla, lo que haremos es grabarlo en un fichero. Es decir, que cada línea corresponderá a la posición de todos los agentes en cada instante determinado (respetando los horarios que se encuentran en el fichero). Esto tiene las siguientes ventajas:

- Nos deshacemos del fichero que contiene los horarios ya que tenemos el registro total del día en un fichero
- Podemos aplicar cálculos sobre el fichero registrado en este último paso (sin este fichero, por ejemplo, no podríamos hacer el mapa de calor dinámico).

- **Generación del mapa de calor dinámico.** Esta parte se ha desarrollado en *C++*. Este mapa de calor se realizará a partir de la traza obtenida en el paso anterior. Este fichero pasará por un tratamiento el cual consistirá en línea por línea analizando el estado y la distribución de cada uno de los puntos. Como el mapa de calor consiste en marcar las zonas de mayor congestión necesitamos una manera de poder observar que zonas son las que están más y menos congestionadas.

La idea, es encontrar un punto de referencia para entender la congestión de cada lugar, el cual no varía respecto al tiempo y mirar cuantos y qué puntos se encuentran en cada zona del punto de referencia. Por esta razón, se realizará una especie de cuadrícula estática encima del mapa para observar en cada momento la situación de todos los agentes dentro de ésta.

Por ejemplo, en la Figura 5 tenemos esta idea implementada donde cada sector tiene un color asignado en función de la concentración de agentes que existe. En esta Figura, nos podemos fijar que en sectores con diferentes concentraciones tendrán diferentes colores (observar tabla 3 para entender la escala de colores) y, los que tengan la misma concentración, el mismo color (cuando están solos tienen un color, cuando son dos otro y así hasta alcanzar un máximo donde a partir de éste tienen el mismo color).

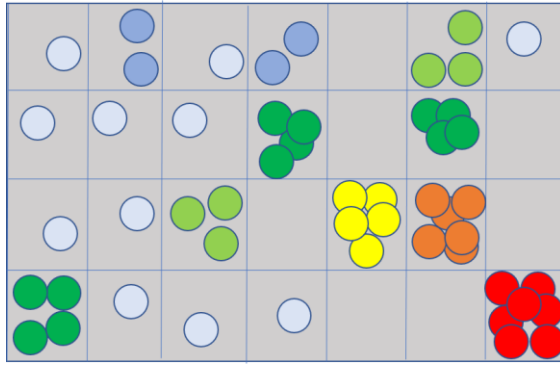


Fig. 5. Clasificación del mapa de calor

Este podría ser ejemplo de un estado real de la ciudad reducido. La tabla de colores aplicada anteriormente de forma reducida es la que podemos ver a continuación, la cual es la que se ha utilizado en la visualización del proyecto:

Color	Hexadecimal	Cantidad(agentes)	Tonalidad
Azul	#00FFFF	1	
Verde	#00FFBF	< 10	
Verde	#00FF80	< 20	
Verde	#00FF40	< 30	
Verde	#00FF00	< 40	
Amarillo	#FFFF00	< 50	
Naranja	#FFBF00	< 70	
Naranja	#FF8000	< 100	
Rojo	#FF4000	< 200	
Granate	#FF0000	> 200	

Tabla 3. Colores según la congestión

- **Generación del cálculo del tamaño de cada línea de la traza final.** Aquí, la única cosa que realizaremos será apuntar el tamaño de cada línea en bytes en un fichero el cuál hará uso nuestro módulo de visualización. Esto, como ya se ha mencionado, tendrá el objetivo de ayudar al módulo de visualización, haciendo que no sea necesario el cargarse todo el fichero. Esto, ha sido desarrollado en *Bash*.

Y, en último lugar, el módulo de visualización, como se inicialmente pensó, es simplemente la lectura de las trazas previamente calculadas mediante *Javascript* y, la idea del marco de simulación. Además, gracias a la idea del fichero de los tamaños de línea, podemos reducir con creces el coste en memoria, por esta razón es posible la escalabilidad en este proyecto. Para ilustrar esta idea, una traza generada puede ocupar alrededor de 3GB y, debido a esto, en memoria solamente será necesario cargarse 2MB, lo que implica una mejora notable.

7. Resultados

En este apartado se mostrarán los resultados obtenidos después de todo el desarrollo del proyecto. A modo de recordatorio, nuestro proyecto es mostrar de forma visual todo el movimiento de la ciudad indicando sus zonas de mayor congestión mediante un mapa de calor.

A modo de demostración de la importancia que tiene este mapa de calor, primero observamos como sería el mapa sin indicar las congestiones.



Fig. 6. Visualización sin mapa de calor

Como se puede observar en la anterior figura, podemos observar donde se encuentran los agentes, pero esta situación no nos dice demasiado. Por esta razón, tras ver que este mapa no nos revelaba demasiada información, pasamos a la implementación del mapa de calor tal y como se puede observar en la siguiente figura.



Fig. 7. Visualización con mapa de calor

Estos resultados, nos hacen ver la importancia de la herramienta visual para observar el tráfico y, en caso de ofrecer esto al ciudadano, se podrían mejorar tiempos de trayecto al conocer las congestiones, zonas *fog* e, incluso, para ayudar a la policía al control de masas de gente. Gracias a que esta visualización está pensada para que el usuario pueda interactuar con ella, se añadió la opción de poder aumentar o disminuir el tamaño de los agentes. Por ejemplo, en la siguiente figura se puede observar la visualización con el tamaño de los agentes aumentado.



Fig. 8. Visualización con el tamaño de los agentes aumentados

Si nos fijamos, a simple vista ya se podrá observar donde se encuentran las concentraciones, pero, en caso de querer reducir el tamaño, en la siguiente figura observaremos como queda la visualización.



Fig. 9. Visualización con el tamaño de los agentes reducidos

Esta interacción hará que el usuario tenga una mayor sensación de control sobre la visualización que tenga delante, haciendo así un entorno más amigable. Además, este tipo de interacción nos permitirá adaptarnos a los gustos del usuario. Debido a que esta interacción está enfocada al usuario, ofreceremos esta interacción en la propia interfaz donde reside la propia visualización mediante botones. Y, además, en la parte superior derecha

tendremos la hora para poder saber a qué instante nos encontramos.



Fig. 10. Interfaz con hora y botones de interacción con la visualización

8. Trabajo futuro

Hace falta aclarar que, aunque el trabajo se planteó como un resultado final, es necesario pensar a largo plazo para su continuo desarrollo y mejora en todos los aspectos. Todos estos aspectos lo aclararemos mediante una lista y así entender y tener una visión más generalizada de lo que podría llegarse a ver en este proyecto en un futuro:

- La posibilidad de trasladarlo a otro entorno.
- Mejorar la cantidad y calidad de la información.
- Mejorar la interacción con la simulación.
- Posibilidad de implementación de gráficos explicativos de lo que ocurre en cada momento de la simulación para un mejor entendimiento.

9. Conclusiones

Cuando este proyecto se me presentó en el CRAAX y me explicaron cuál era la idea inicial, pensé que el trabajo no sería realmente muy complicado, pero, en cuanto entré a revisar tecnologías y formas de realizarlo me di cuenta de la dimensión del problema.

La realidad es que cuando pensé que era la tecnología perfecta, la gran cantidad de *APIs* y librerías visuales diferentes me hacían pasar por cada una de ellas revisando cuál satisfacía mis especificaciones ya que cada una de ellas tenían características diferentes. Lo que me encontré fue, que ninguna de ellas podía satisfacerlas, por eso se tuvo que volver al diseño y reestructurar toda la idea inicial.

En cuanto el diseño se reestructuró la progresión fue lineal, es decir, no encontramos ningún tipo de problema adicional causado por la tecnología, lo cual fue un gran alivio ya que existieron muchos problemas en relación con ese aspecto. Es cierto que el desarrollo de algunos algoritmos fue complicado, pero gracias al diseño bien estructurado no

fueron un gran problema ya que la idea era fija y muy focalizada.

Referencias

- [1] Patrones de diseño en proyectos de *Javascript*.
<https://addyosmani.com/largescalejavascript/>
- [2] Fuente de la distribución geográfica de barrios en Vilanova i la Geltrú.
<http://www.vilanova.cat/html/tema/barris/descbarris.html>
- [3] Cantidad de personas por edad en Vilanova i la Geltrú.
<https://www.idescat.cat/pub/?id=pmh&n=1180&geo=mun:083073&lang=es>
- [4] Cantidad de personas por edad en Vilanova i la Geltrú.
<https://www.idescat.cat/pub/?id=pmh&n=1180&geo=mun:083073&lang=es>
- [5] Tasa de paro en Vilanova i la Geltrú.
<https://www.datosmacro.com/paro/espana/municipios/cataluna/barcelona/vilanova-i-la-geltru>
- [6] Repositorio con el código implementado del proyecto.
<https://github.com/EpsilonZ/CityVisualizer>