

# **Aplicaciones de Ciencias de Datos y Tecnologías Inteligentes**

**Transporte, Localización y Patrullaje:  
Trabajo Teórico-Práctico**

**Aplicación: “WhatTappa”**

**Antonio Manuel Milán Jiménez**  
**[antoniomj@correo.ugr.es](mailto:antoniomj@correo.ugr.es)**

|   |           |
|---|-----------|
| <b>Introducción</b>   | <b>3</b>  |
| <b>Aplicaciones de rutas de tapas</b>                                 | <b>4</b>  |
| <b>Sistema inteligente de rutas de tapas basado en Soft Computing</b> | <b>5</b>  |
| Componentes   | 5         |
| Funcionamiento  | 6         |
| Uso de lógica difusa  | 8         |
| Variable de entrada: Movimiento del usuario                           | 8         |
| Variable de entrada: Concurrencia del local                           | 9         |
| Uso de metaheurísticas  | 10        |
| <b>Manual de usuario</b>  | <b>11</b> |
| <b>Conclusión</b>   | <b>13</b> |
| <b>Referencias</b>  | <b>13</b> |

# Introducción

Uno de los mayores atractivos turísticos de España, y más aún en Granada si cabe, es su gastronomía. Miles e incluso millones de personas se sienten atraídas cada año a visitar nuestro país por su famosa cocina. Concretamente en nuestra región, las “tapas” han tomado una relevancia mundial, siendo la opción preferida por españoles y extranjeros a la hora de comer.

Las rutas de tapas se han convertido en toda una atracción turística: probar diferentes sitios de “tapeo” a lo largo de la ciudad, una forma interesante de comer y recorrer sus calles. Dada esta atracción turística, han surgido muchos sitios de tapas, mejores y peores. No es de extrañar que se dude sobre a qué sitios ir dada la amplia oferta que se ofrece.

Lo más habitual es recurrir a aplicaciones tan famosas como “TripAdvisor” [1], que ofrecen los mejores sitios para comer en función de la puntuación de los propios usuarios. Sin embargo, dada ya la relevancia de esta aplicación, es muy habitual descubrir que los 3 o 4 mejores sitios mostrados en la aplicación están completamente llenos, siendo imposible comer y “descuadrando” la ruta que se había pensado en un primer momento. Entonces el usuario normalmente vuelve a sacar el móvil para ver los siguientes mejores sitios, dudando ya de cómo de lejos estarán o si volverán a llenarse.

Granada

Publicar
Viajes
Buzón
Perfil
Unirse
Buscar

Muro de viajes: Granada
Hoteles
Qué hacer
**Restaurantes**
Vuelos
Alquiler Vacacional
Paquetes de Vacaciones
Coches de alquiler
...

Europa > España > Andalucía > Provincia de Granada > Granada > Restaurantes en Granada
Mejores restaurantes en Granada, Provincia de Granada

## Restaurantes en Granada

**Reserva:**

30/5/2019
20:00
2 clientes
Encuentra un restaurante
Borrar búsqueda

[Ver mapa](#)

**Tus selecciones**
[Borrar todo](#)

- Bares y pubs
- Española

**Tipo de establecimiento**

- ☒ Bares y pubs
- ☐ Restaurantes
- ☐ Tentempiés

Más ▾

**Reservas**

- ☐ Reservas online
- ☐ Ofertas de restaurantes
- ☐ Disponibles para esta noche

**Cocinas y platos**

- ☒ Española
- ☐ Café

**Ordenar por:**
Disponibilidad
**Relevancia**
Clasificación
Nombre

Hemos encontrado interesantes resultados, pero algunos están fuera de Granada. Mostrando resultados en ciudades cercanas.  
Limitar búsqueda a Granada.

**Rosario Varela**

4.5
974 opiniones
#1 de 7 resultados
€€ - €€€, Bar, Mediterránea, Europea, Española, Opciones vegetarianas, Opciones ...

"Taperío" 20/05/2019  
"TAPAS Y VERMOUTH MUY ORIGINAL. TRA..." 17/04/2019

**la jarana** Almuñécar a 0.9 km

4.5
42 opiniones
#2 de 7 resultados
€. Mediterránea, Española

"Vacaciones" 12/10/2018  
"Buen sitio" 02/10/2018

**Mamut Bola de Oro**

4.5
77 opiniones
#3 de 7 resultados
€€, €€€, Española Fusión, Vinoteca

Frente a este problema surge “WhatTappa”. A partir de los mejores sitios de tapas cercanos al usuario, esta aplicación para Android mostrará al usuario la ruta de tapas más eficiente. No solo se considerará la distancia entre los sitios para idear la ruta, sino que también tendrá en cuenta cómo de lleno está el sitio en tiempo real, gracias a la información que aporten los propios usuarios en ese momento. De esta forma, la aplicación hará por distribuir a los usuarios por los diferentes lugares, propiciando que prueben los mejores sitios pero sin que se generen esos “cuellos de botella”.

Por lo tanto en este trabajo se presentará el diseño de esta aplicación, un sistema inteligente que incorporará técnicas de Soft Computing para trazar rutas de tapas eficientes y para ayudar al usuario a considerar cómo de lleno está el lugar con un lenguaje más cercano. También se mostrará en este trabajo un estudio de mercado sobre este tipo de aplicaciones, el funcionamiento de la aplicación y su uso.

## **Aplicaciones de rutas de tapas**

A continuación se detallan algunas de la aplicaciones ya presentes en este mercado:

- TripAdvisor [1]: Sitio web que ofrece diversas recomendaciones de hoteles, viajes y restaurantes a sus usuarios en función de la propia valoración de usuarios anteriores. Concretamente para este mercado, es capaz de ofrecer los mejores restaurantes/bares más cercanos al usuario.
- Rutappa [2]: Aplicación móvil que da soporte a la organización de rutas de tapas por los propios usuarios. Incorpora la valoración de las rutas por los usuarios y muestra los locales ordenados en función de su distancia al usuario.
- Tapitasapp [3]: Aplicación móvil que ofrece información sobre los locales de tapas más cercanos al usuario. Se muestran las tapas así como su valoración. Además, permite que los usuarios creen sus rutas personalizadas de tapas y las compartan con el resto de usuarios.

Aunque todas estas aplicaciones muestren información sobre los locales más cercanos, ninguna de ellas traza una ruta como tal para hacer más eficiente el recorrido. Si bien en un área pequeña de recorrido no es tan importante, sí que puede ser relevante si el usuario va a realizar un recorrido por locales más distantes.

Más allá de esto, ninguna de estas aplicaciones ofrece información en tiempo real sobre cómo de concurrido está el local, un aspecto muy importante a la hora de elegir a qué sitio ir.

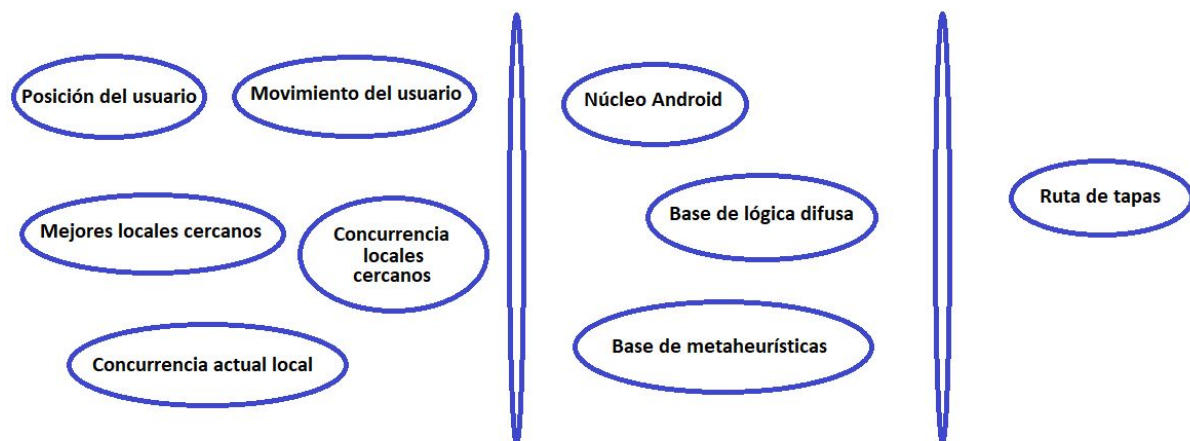
# Sistema inteligente de rutas de tapas basado en Soft Computing

Como se ha introducido anteriormente, el objetivo de este trabajo es diseñar un sistema inteligente capaz de ofrecer las mejores rutas de tapas en base a dónde se encuentran los locales y cómo de concurridos están.

Para poder alcanzar este objetivo, el sistema debe incorporar concretamente dos técnicas de Soft Computing. El uso de lógica difusa para que la comunicación con el usuario sea más cercana y el uso de metaheurísticas para trazar un camino eficiente entre los diferentes locales, considerando no solo la distancia sino también cómo de llenos están.

## Componentes

A continuación se detalla el esquema de las componentes del sistema:

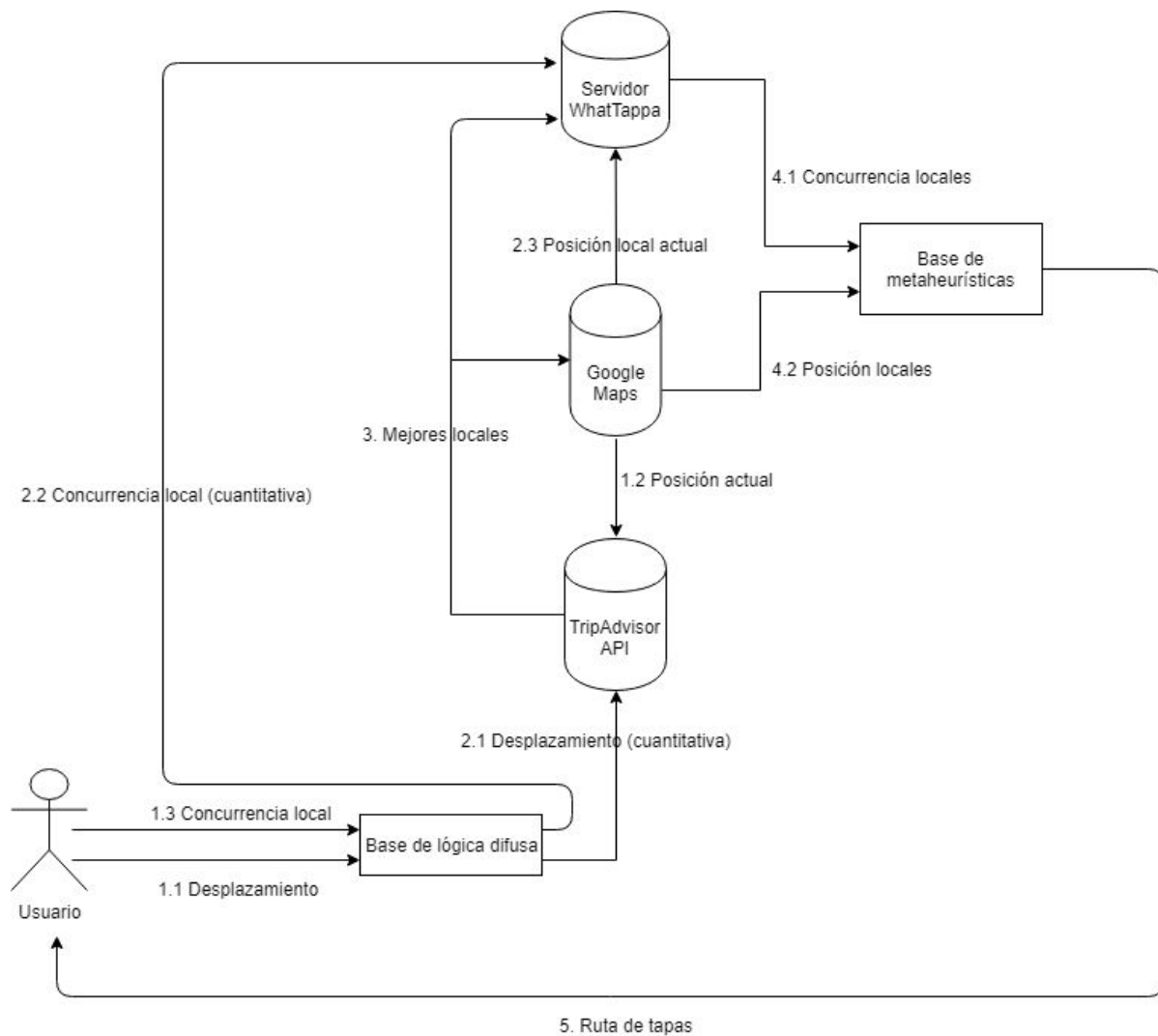


- Subsistema de Entrada: En este subsistema se procesa la información de entrada referente a la posición del usuario, cuánto está dispuesto a moverse el usuario, los mejores locales cercanos y cómo de concurridos están los locales.
  - Posición del usuario: Coordenadas proporcionadas por la API de Google Maps [4] para conocer dónde está el usuario en cada momento.
  - Movimiento del usuario: Información cualitativa introducida por el usuario de cuánto está dispuesto a moverse desde su posición inicial.
  - Mejores locales cercanos: En base a la posición inicial del usuario y su movimiento, la API de TripAdvisor [5] ofrece los 7 mejores locales de tapas en ese radio. Además, sus coordenadas son nuevamente proporcionadas por Google Maps.

- Concurrencia de los locales seleccionados: Información cualitativa de usuarios previos, proporcionada a través del servidor de la aplicación, referente a cómo de llenos están los locales que compondrán la ruta.
- Concurrencia actual del local: Información cualitativa introducida por el usuario indicando cómo de lleno está el local en el que se encuentra.
- Subsistema Central: Sobre este subsistema sucede el funcionamiento principal de la aplicación apoyado en un núcleo Android, una base de lógica difusa, una base de metaheurísticas y la geolocalización:.
  - Núcleo Android: Núcleo del sistema operativo Android encargado de las diferentes peticiones de funcionamiento de la aplicación.
  - Base de lógica difusa: Se encarga del tratamiento de la información cualitativa relativa al movimiento y concurrencia, introducida por el usuario, para que pueda trabajar la aplicación internamente sobre ella.
  - Base de metaheurísticas: Se encarga del cálculo de un recorrido eficiente entre los diferentes locales considerando sus distancias gracias a Google Maps y los últimos niveles de concurrencia proporcionados.
- Subsistema de Salida: En este subsistema se muestra sobre un mapa al usuario la ruta de tapas calculada que debe seguir. A través de Google Maps el usuario sabe dónde se encuentra en cada momento. Además, la ruta se puede actualizar en función de nueva información sobre concurrencia que proporcionen otros usuarios.

## Funcionamiento

A continuación se detalla el esquema de funcionamiento del sistema:



Inicialmente, el sistema solicita al usuario que indique cuánto está dispuesto a desplazarse dada su ubicación actual. La base lógica difusa se encarga de convertir esta información cualitativa a cuantitativa mediante un proceso de “defuzzificación”. Así, ya esta información se utiliza a modo de filtro la API de TripAdvisor [5] para obtener los mejores locales de tapas en ese radio.

A continuación, para los locales que han sido seleccionados se obtiene sus posiciones gracias a la API de Google Maps [4] y se obtiene también cómo de concurridos están en base a la información proporcionada por usuarios anteriores en los últimos 30 minutos, información ya convertida a cuantitativa.

Una vez realizado esto, la base de metaheurísticas se encarga de calcular un recorrido eficiente entre los locales, es decir, la ruta de tapas en sí. Resuelve este problema como un “viajante de comercio” considerando tanto la distancia entre los locales proporcionada por Google Maps como sus concurrencias proporcionadas por los usuarios.

Ya determinada la ruta, se muestra al usuario utilizando nuevamente la API de Google Maps [4]. Además, se muestra en cada momento dónde se encuentra el usuario en el recorrido. Es posible que esta ruta se recalculé en función de nueva información sobre concurrencia que aporten otros usuarios.

Respecto a esto, adicionalmente el usuario puede introducir de manera cualitativa cómo de lleno está el local. Así, la base de lógica difusa realiza un proceso de ‘defuzzicación’ sobre esta información que se almacena en el servidor de la aplicación junto con el local en el que se encuentra, gracias a la geolocalización.

## Uso de lógica difusa

En este apartado se detalla el tratamiento de la información cualitativa introducida por el usuario por parte de la base de lógica difusa.

### Variable de entrada: Movimiento del usuario

Esta variable está relacionada con cuánto está dispuesto a moverse el usuario dada su posición inicial para saber en qué radio considerar los locales de tapas. El usuario puede escoger entre las siguientes etiquetas cualitativas para hacer más cercano el lenguaje al usuario: [Muy poca, Poca, Normal, Bastante, No importa]. Aquí se muestran las funciones de pertenencia:



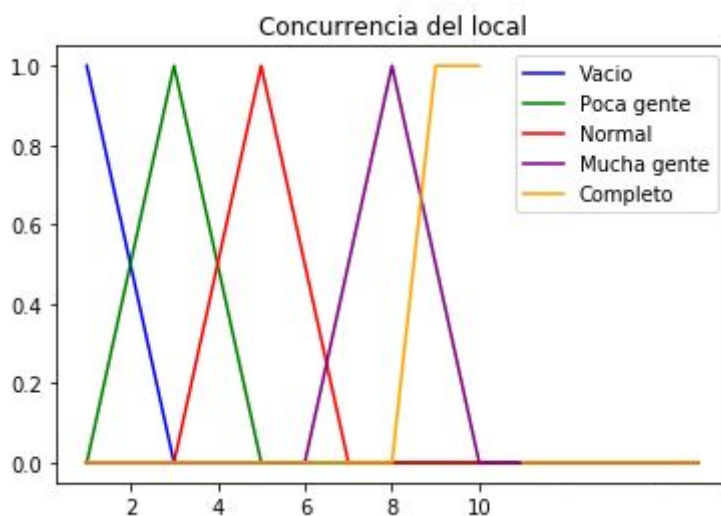
Como proceso de ‘defuzzicación’ se tomaría la máxima pertenencia con mayor valor en el eje X, es decir, en distancia. De esta forma se delimita el radio máximo en el que está dispuesto a desplazarse el usuario. Se tendría entonces:



| Valor cualitativo | Valor cuantitativo |
|-------------------|--------------------|
| Muy poca          | 50 metros          |
| Poca              | 150 metros         |
| Normal            | 400 metros         |
| Bastante          | 600 metros         |
| No importa        | 900 metros         |

### Variable de entrada: Concurrencia del local

Esta variable está relacionada con la valoración del usuario de cómo de lleno está el local en el que se encuentra. De igual forma que en el caso anterior, se trata de información cualitativa para hacer más cercano el uso de la aplicación al usuario. Las etiquetas lingüísticas son: [Vacío, Poca gente, Normal, Mucha gente, Completo]. Se muestran ahora las funciones de pertenencia:



Respecto al proceso de 'defuzzificación' se tomaría el valor de máxima pertenencia:

| Valor cualitativo | Valor cuantitativo |
|-------------------|--------------------|
| Vacío             | 1                  |
| Poca gente        | 3                  |
| Normal            | 5                  |
| Mucha gente       | 8                  |
| Completo          | 10                 |

Dado que para un mismo local se pueden tener diversas valoraciones de su concurrencia por parte de distintos usuarios, se realiza una media de las valoraciones en los últimos 30 minutos como consideración final de su concurrencia.

## Uso de metaheurísticas

Para la creación en sí de las rutas, es necesario el uso de metaheurísticas para poder calcular un recorrido eficiente en base a las distancias entre los locales y cómo de llenos están.

Se afronta este problema como una variación del problema de “viajante de comercio” en el que como posible solución se tiene una permutación de los diferentes locales representando el recorrido que se hace entre ellos; pero que también se tendrá que tener en cuenta la concurrencia de cada uno de ellos.

A modo de ejemplo, si se tienen los locales  $L_1$ ,  $L_2$ ,  $L_3$ ,  $L_4$  y  $L_5$ ; una posible solución sería:  $[L_2, L_4, L_1, L_5, L_3]$  indicando que el recorrido entre los locales se realizaría de la forma  $S \rightarrow L_2 \rightarrow L_4 \rightarrow L_1 \rightarrow L_5 \rightarrow L_3$  siendo  $S$  la posición inicial del usuario. Para esta solución, la distancia total recorrida sería la suma de las distancias entre  $S$  y  $L_2$ ,  $L_2$  y  $L_4$ ,  $L_4$  y  $L_1$ ,  $L_1$  y  $L_5$ ,  $L_5$  y  $L_3$ . Esta sería la medida que se intentaría minimizar en el caso de un problema clásico de “viajante de comercio”, realizando diferentes permutaciones sobre la solución con el fin de poder recorrer los diferentes locales en la menor distancia posible.

Sin embargo, para considerar además la concurrencia en las posibles soluciones y sus valoraciones, hay que tener en cuenta qué concurrencia actual tiene cada local y cuándo se visitaría. Por ejemplo, si un local en el momento de construir la ruta tiene una concurrencia muy alta, no sería recomendable que fuese de los primeros en visitarse pues propiciaría el “cuello de botella” mencionado en anteriores secciones. De igual forma, si un local tiene en ese momento poca gente, es interesante que sea de los primeros en visitarse para distribuir mejor la cantidad de gente en cada sitio. Por lo tanto, una forma de proceder con esto es que si en la solución se tiene  $L_3 \rightarrow L_2$ , la distancia considerada  $D_{3-2}$  se multiplicaría por la concurrencia de  $L_2$  y por un valor del vector  $[1, 0.8, 0.6, 0.4, 0.2]$  relativo a la posición en la

que es visitado el local, dando más peso así a la concurrencia cuanto antes vaya a ser visitado el sitio.

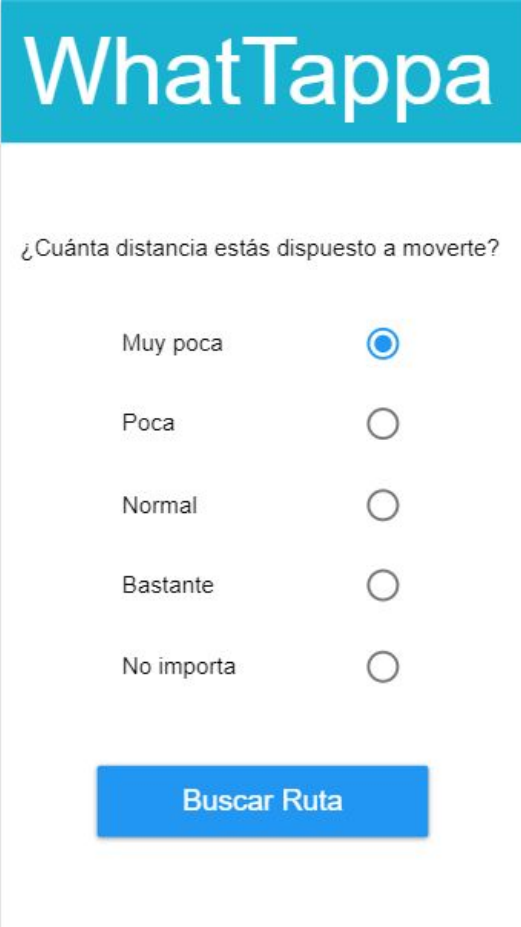
Por ejemplo, para la solución  $S \rightarrow L_2 \rightarrow L_4 \rightarrow L_1 \rightarrow L_5 \rightarrow L_3$ , su valoración final sería:

$$D_{S-L_2} * Conc_{L_2} * 1 + D_{L_2-L_4} * Conc_{L_4} * 0.8 + D_{L_4-L_1} * Conc_{L_1} * 0.6 + D_{L_1-L_5} * Conc_{L_5} * 0.4 + D_{L_5-L_3} * Conc_{L_3} * 0.2$$

Ya entonces, esta valoración de las soluciones se trataría siempre de minimizar; empleando un algoritmo genético de 100 épocas para la búsqueda de soluciones hasta obtener finalmente una buena ruta de tapas.

## Manual de usuario

Respecto a cómo utilizar esta aplicación, el usuario simplemente tendrá que indicar cuánta distancia se movería eligiendo entre las opciones de [Muy poca, Poca, Normal, Bastante, No importa]:



The screenshot shows the 'WhatTappa' application interface. At the top is a blue header with the text 'WhatTappa' in white. Below the header is a white box containing the question '¿Cuánta distancia estás dispuesto a moverte?'. Underneath this question are five radio button options: 'Muy poca', 'Poca', 'Normal', 'Bastante', and 'No importa'. The 'Muy poca' option is selected, indicated by a blue dot in the center of its radio button. At the bottom of the white box is a blue button with the text 'Buscar Ruta' in white.

Automáticamente se calcularía la ruta, mostrándola sobre un mapa junto con la posición del usuario:

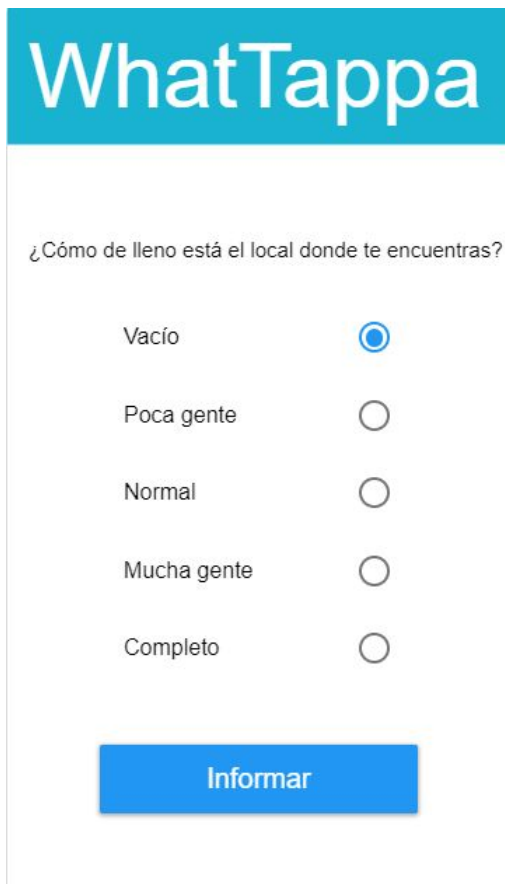
# WhatTappa



¿Cómo de lleno está el local?

Salir

Adicionalmente, el usuario podría especificar cómo de lleno está el local en el que se encuentra eligiendo entre las opciones de [Vacío, Poca gente, Normal, Mucha gente, Completo]:



WhatTappa

¿Cómo de lleno está el local donde te encuentras?

Vacío ☒

Poca gente ☐

Normal ☐

Mucha gente ☐

Completo ☐

Informar

## Conclusión

En este trabajo se ha presentado una aplicación para móviles Android que proporciona al usuario una ruta personalizada de tapas. Considera el desplazamiento del usuario y se diferencia del resto de aplicaciones en considerar cómo de llenos están los diferentes locales de tapas, algo fundamental a la hora de elegir a cuál ir. Se consiguen estas funcionalidades gracias al Soft Computing, utilizando lógica difusa y metaheurísticas.

## Referencias

1. TripAdvisor: Sitio web de recomendación de hoteles, restaurantes y vuelos. <https://www.tripadvisor.es/>
2. Rutappa: Aplicación móvil de soporte de rutas de tapas. <http://rutappa.es/>
3. Tapitasapp: Aplicación móvil que ofrece información sobre las tapas más cercanos. <https://tapitasapp.com/>
4. Google Maps API: <https://cloud.google.com/maps-platform/?hl=es>
5. TripAdvisor API: <https://developer-tripadvisor.com/content-api/>