

YourPlacesBot - A Telegram Bot

David Quesada López y Mateo García Fuentes

GRADO EN INGENIERÍA INFORMÁTICA

FACULTAD DE INGENIERÍA INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID



UNIVERSIDAD  
**COMPLUTENSE**  
MADRID

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

Director: Carlos Gregorio Rodríguez

17 de mayo de 2017



UNIVERSIDAD  
COMPLUTENSE  
MADRID

**AUTORIZACIÓN PARA LA DIFUSIÓN DEL TRABAJO FIN DE GRADO Y SU DEPÓSITO EN EL REPOSITORIO INSTITUCIONAL E-PRINTS COMPLUTENSE**

Los abajo firmantes, alumno/s y tutor/es del Trabajo Fin de Grado (TFG) en el Grado en .....de la Facultad de ....., autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el Trabajo Fin de Grado (TF) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Periodo de embargo (opcional):

- ☐ 6 meses  
☐ 12 meses

TÍTULO del TFG: .....

Curso académico: 20.... / 20....

Nombre del Alumno/s:

.....  
.....

Tutor/es del TFG y departamento al que pertenece:

.....  
.....  
.....

Firma del alumno/s

Firma del tutor/es

## **Agradecimientos**

Gracias a Nick Lee (<https://github.com/nickoala>) por desarrollar telepot, un framework de Python para API de Telegram Bot y desarrollarlo bajo una licencia MIT.

# Índice general

Índice de figuras	v
Índice de abreviaturas	vi
Resumen	vii
Abstract	viii
<b>Capítulo 1: Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	1
1.3. Estructura del documento . . . . .	2
<b>Capítulo 2: Infraestructura para bots - Telepot</b>	<b>4</b>
2.1. Plataforma de bots de Telegram . . . . .	4
2.2. Telepot . . . . .	4
<b>Capítulo 3: Experimentación previa</b>	<b>6</b>
<b>Capítulo 7: Conclusiones y trabajo futuro</b>	<b>7</b>
<b>Apéndice</b>	<b>8</b>
<b>Bibliografía</b>	<b>9</b>
<b>Anexo</b>	<b>10</b>
<b>Glosario</b>	<b>11</b>

## Índice de figuras

## Índice de abreviaturas

# Resumen

Una de los principales atractivos de Telegram es su plataforma para bots. Los usuarios pueden crear sus propios bots y ponerlos en funcionamiento para que sean accesibles a todos los clientes de Telegram sin coste alguno para el desarrollador o para el consumidor. Estos bots proporcionan en su mayoría información, juegos o utilidades dentro de un chat, y aumentan en gran medida la funcionalidad de Telegram.

A la hora de que un usuario interactúe con un bot, es especialmente interesante que el servicio que se le preste pueda depender de su ubicación geográfica y pueda tener un componente social. Por ello, este proyecto tiene como objetivo informar al usuario sobre qué establecimientos cercanos hay en base a su localización, ofreciendo la posibilidad de encontrarlos fácilmente y de ver datos proporcionados por otros usuarios sobre estos.

Palabras clave: bot, Telegram, geolocalización, noSQL, mapas de calor.

# Abstract

One of the main features of Telegram is its bot support. Users can create their own bots and launch them to be available for everyone in Telegram without cost for neither the developer nor the client. These bots mainly offer information, games or utilities inside the chat and they increase greatly Telegrams functionality.

When an user interacts with a bot, it is of special interest that the response of the bot varies depending on the users location and on a social component. That's why this project aims to inform the user about what near by establishments there are depending on his location, offering the possibility to find them easily and to see information of them given by other users.

Keywords: bot, Telegram, geolocation, noSQL, heatmaps.



# Capítulo 1: Introducción

## Motivación

La motivación para crear este bot surgió a partir de unas conversaciones con amigos. Se quejaban de no disponer de una herramienta sencilla para poder ver qué locales hay cercanos a nosotros y así elegir a dónde ir si no conoces el lugar.

Al intentar buscar una solución a este problema lo primero en lo que pensamos fue la aplicación de Google Maps. Google Maps tiene la funcionalidad de encontrar establecimientos cercanos a ti en base a tu localización y a qué buscas, pero el resultado fue que la aplicación es demasiado grande y no es práctico tener que abrirla, navegar entre todas las opciones y funcionalidades que ofrece y ponerte a buscar un local que te guste.

En esta situación, pensamos que un bot de Telegram encajaría bien como solución. Por un lado, Telegram es una aplicación que usamos día a día y un bot resultaría fácil de manejar. Por otro lado, la función de compartir la ubicación de Telegram se adecúa perfectamente a la API de Google Maps para realizar consultas de búsqueda de establecimientos cercanos.

## Objetivos

El objetivo de este proyecto será la implementación de un bot de Telegram que resuelva el problema expuesto anteriormente valiéndose de la ubicación que se le envíe para devolver locales con las características indicadas. Para esto, vamos a establecer una serie de objetivos que el bot debe cumplir:

- Ser capaz de **recibir una localización y devolver una lista de locales cercanos a ella**. Estos locales deberán estar dentro de un

cierto radio de distancia variable y por defecto se buscará que estén abiertos en el instante de la búsqueda.

- Cuando se seleccione un establecimiento, **se devolverá su ubicación** y la información de la que disponga el bot sobre él.
- Poderle dar una puntuación a los establecimientos y ser capaz de enviar fotografías al bot.
- Poder mostrar un **mapa de calor** en torno a tu localización reflejando las ubicaciones más concurridas por otros usuarios

## Estructura del documento

Vamos a estructurar este documento conforme al orden en el que fuimos desarrollando el bot. Para cada versión con nuevas funcionalidades, documentaremos qué hemos cambiado, cómo lo hemos hecho y a qué problemas nos hemos enfrentado.

A continuación mostramos sobre qué va a tratar cada capítulo:

- **Capítulo 2: Infraestructura para bots - Telepot.** En este capítulo introduciremos al lector al desarrollo de bots en Telegram contando cómo es la infraestructura que tienen y la API que ofrece Telegram a los desarrolladores. También hablaremos de qué es y por qué hemos usado el framework de Telepot para nuestro bot en vez de la API nativa de Telegram.
- **Capítulo 3: Experimentación previa.** Esta parte tratará de nuestro primer acercamiento a la creación de bots. Hablamos de cómo empezamos a usar Telepot para codificar bots simples y lo que aprendimos para usar en un futuro en nuestro bot final.
- **Capítulo 4: Primera versión - Establecimientos cercanos.** En este apartado hablamos sobre el desarrollo de la primera versión estable de YourPlacesBot. Esta versión ya cuenta con la funcionalidad y la interfaz básicas y sirve como base sobre la que poder ir implementando mejoras.
- **Capítulo 5: Segunda versión - Base de datos e interacción.** Aquí tratamos el desarrollo de la versión que ya era capaz de realizar toda la interacción entre el bot y el usuario. Es la primera versión que pusimos a funcionar de manera continuada en Telegram.
- **Capítulo 6: Tercera versión - Opciones y Heatmaps.** Este apartado trata sobre la última versión del bot, donde se introducen

opciones sobre el funcionamiento del bot que cada usuario puede variar y mapas de calor en base a tu ubicación.

- **Capítulo 7: Conclusiones y trabajo futuro.** En este último capítulo sacamos conclusiones sobre el desarrollo del bot en Telegram y exponemos las mejoras futuras a implementar en el bot.

# Capítulo 2: Infraestructura para bots - Telepot

## Plataforma de bots de Telegram

El servicio de mensajería Telegram Messenger ofrece una interfaz de programación para bots, llamada Telegram Bot API. Estos bots son aplicaciones de terceros que se ejecutan en Telegram. La forma de interactuar con estos bots es muy variada: mensajes, comandos, archivos, botones, etc. Las características usuales de estos bots son proporcionar notificaciones y noticias personalizadas, juegos o integración de servicios externos a Telegram.

Los mensajes, comandos y peticiones enviados por los usuarios se envían al software que se ejecuta en los servidores. Los servidores de Telegram, que son intermediarios entre nuestro servidor y el usuario, manejan el cifrado y la comunicación. Esta comunicación se realiza a través de una interfaz HTTPS que ofrece una versión simplificada de la API de Telegram.

Los desarrolladores deben crear sus bots a través de @BotFather. Este bot es el padre de todos los bots y proporciona tokens únicos para cada uno. También es necesario para administrar los bots y gestionar sus configuraciones. Además de esto, el @BotFather avisa a los desarrolladores de bots populares (300 solicitudes por minuto o más) si detecta que la tasa de conversión de solicitud/respuesta es muy baja. Está alerta consta de 3 botones de respuesta: Arreglado, Silenciar y Apoyo. Esta última opción abre un nuevo chat con el @BotSupport que ayudará al desarrollador a detectar el problema o averiguar si hay algún problema en los servidores de Telegram.

## Telepot

Telepot es un framework desarrollado por Nick Lee (<https://github.com/nickoala/telepot>) que te ayuda a desarrollar bots de

Telegram usando Python en vez de HTTPS. La funcionalidad básica de Telepot consiste en servir de wrapper de todas las funciones de la API de Telegram para así poder programar el código de los bots en Python.

Además de esto, también ofrece funcionalidades para hacer más cómoda la gestión de servidor de tu bot. Cuando creas un bot, tienes que ir pidiendo a los servidores de Telegram *updates* para que te envíen una lista con los mensajes que le han sido enviados. Telepot te proporciona métodos para que no tengas que preocuparte de montar toda una estructura que gestione su recepción. Telepot se encarga de crear hilos de ejecución por cada chat nuevo que se abre con el bot y de pararlos cuando estos llevan demasiado tiempo sin realizar ninguna tarea.

## Capítulo 3: Experimentación previa

## Capítulo 7: Conclusiones y trabajo futuro

## Apéndice



# Bibliografia

- [1] LaTeX -><http://texdoc.net/texmf-dist/doc/latex/memoir/memman.pdf>
- [2] Python -><https://docs.python.org/2.7/>
- [3] API Google Maps Docs  
-><http://googlemaps.github.io/google-maps-services-python/docs/2.4.5/>
- [4] Telegram Bots -><https://core.telegram.org/bots/api>
- [5] API Telepot -><http://telepot.readthedocs.io/en/latest/>

## Anexo

## Glosario