**Nombres de los participantes:**

Farfan Perdomo Jorge

Martínez Vargas Zaira Carolina

Alaniz Fabian Carlos Felix

Roman Chitala Luis Ernesto

1. **Nombre del proyecto**

Optimización de traslados hacia clínicas de emergencia.

Nombre Corto: Movilízat[e](https://www.ocu.org/movilizate-ocu) Salud

1. **Eje rector del proyecto**

Salud y movilidad debido a la deficiencia que se encuentra en el sector salud del país al atendernos en caso de una eventualidad.

1. **Planteamiento del problema**

Ante una situación repentina que requiere atención médica, no se cuenta con una herramienta más allá de las aplicaciones de mapas, que nos permita encontrar de manera ágil los servicios médicos de emergencia disponibles a nuestro alrededor así como estimar los tiempos de traslado y espera en los mismos.

La herramienta que se está desarrollando funcionará dentro de la ZMG para encontrar un hospital de forma ágil en el cual puedan atender, ya que algunas veces se desconoce la existencia de clínicas y hospitales cerca y si hay sobrecupo en estas.

La ventaja de este proyecto es que se tendrá la certeza de que te podrás atender, también se podrá decidir dónde atenderse y proporcionará una variedad de opciones. Es importante mencionar que ya no se tendrá la necesidad de esperar en la línea de 911 porque esto puede generar un considerable tiempo de espera debido a la gran cantidad de llamadas falsas.

Ignacio Pérez Vega (2019). Enfermeras y doctores del IMSS se quejan de falta de personal, carencias de materiales y aparatos médicos. 2020, de Universidad de Guadalajara. Sitio web:

<http://udgtv.com/noticias/enfermeras-doctoras-imss-quejan-falta-personal-carencias-materiales-aparatos-medicos/>

1. **Descripción de la solución a la problemática detectada.**

Se planea utilizar las herramientas de Machine Learning para emplear un algoritmo que nos ayude a estimar el tiempo de traslado y calcular la ruta óptima hacia los distintos servicios de emergencias médicas alrededor de un punto geográfico escogido de manera arbitraria y que el algoritmo aprenda en base a entrenamiento con experiencias pasadas y mejore sus resultados.

Nuestro proyecto se divide en dos partes:

**1-. Búsqueda de Hospitales Cercanos**

Para delimitar el espacio de búsqueda se utilizará un algoritmo de clustering para delimitar el espacio de búsqueda y poder optimizar el consumo de datos de la API y la eficiencia del mismo.

**2-. Saturación del Hospital**

Una vez encontrado el hospital, se utilizará un algoritmo de Red Neuronal, Perceptrón Multicapa o Adalina, para poder dar la mejor opción en tiempo de traslado, espera y porcentaje de ocupación.

1. **Hipótesis**

Se puede mejorar la distribución de demanda de servicios de emergencia en tiempo casi real así como hacer más eficiente el traslado al contar con información en tiempo real de manera accesible y compacta. Esto se logrará a través de una app que utilice la geolocalización del dispositivo y con base en ello muestre una lista de las opciones más viables.

1. **Metodología** 
   1. Descripción de los datos:

Se cuentan con datasets de los recursos de salud de los hospitales en la ZMG (e.g. Número de médicos por hospital, camas disponibles, consultorios), Registros de urgencias, Egresos hospitalarios, coordenadas geográficas de los hospitales dentro de la ZMG (estos datos incluyen Hospitales y Clínicas).

* 1. Descripción del modelo a utilizar

Posiblemente se utilizará una API de mapas y un algoritmo de Machine Learning para estimar y hacer más eficiente la delimitación de traslados y posteriormente un modelo de Machine Learning para que la herramienta desarrolle y mejore las estimaciones. Se proponen redes neuronales para estimar tiempo de espera y mejorar las recomendaciones de hospitales.

* 1. Delimitaciones

De momento hacen falta hospitales sobre todo privados así como datos que nos indiquen el número de personal médico disponible y la cantidad de pacientes que ingresan en un determinado tiempo. Necesitamos esto para estimar el tiempo de espera. También feedback para entrenar y mejorar el desempeño de las estimaciones, sobre todo en los tiempos de espera.

(Más metodología)

1.- El usuario final accede a la aplicación en dos posibles casos:

a) Uso previo de la aplicación (llenado de SAMPLER)

b) Emergencia o afección considerada grave.

2.- El primer dato que la aplicación tomará serán las coordenadas actuales del usuario.

3.- Contando con las coordenadas, el modelo realizará el proceso de delimitación geográfica mediante la utilización de un algoritmo de machine learning.

4.- Una vez determinado el grupo de las unidades más próximas, el modelo evaluará la ocupación y tiempo de espera el cual tendrá diversos casos(obtenidos de su previo entrenamiento):

a) Si las variables mencionadas anteriormente exceden el parámetro máximo de traslado o tiempo de espera, el modeló realizará una segunda evaluación con un segundo rango de distancias, hasta agotar los mismos.

I)En caso de encontrar al grupo de unidades médicas preciso, el modelo procederá con el siguiente paso.

b) Si el modelo determina a un grupo de unidades médicas cercanas como aptas, procederá al siguiente paso.

5.- El modelo entonces generará la consulta y petición de la API de maps, para confirmar la veracidad con datos en tiempo real respecto a tráfico.

6.- Una vez confirmada la mejor unidad, se enlistará junto a sus precedentes en rango de menor tiempo total hasta la atención.

7.- El usuario final llegará a su destino y logrará ser atendido adecuadamente.

8.- Se pedirá un feedback al usuario final para reforzar el algoritmo de machine learning.

1. **Marco Teórico**

Existen aplicaciones disponibles para encontrar tu hospital más cercano cuyas descripciones mencionan características muy similares a lo que se propone en este proyecto. Probamos algunas disponibles en la PlayStore para dispositivos Android.

Entre las que se probaron tenemos a:

Hospital Finder by Darshan Institute of Engineering & Technology

Hospital Finder by ArogyaOnline Healthcare

Nearby Hospitals by karthikdeva

Nearby Near Me Hospital by King Coder

Por ejemplo y de acuerdo a la descripción de la app:

Neraby Near Me Hospital da una lista completa de hospitales en un rango de 10 km.

Nearby Hospitals proporciona una lista de hospitales basados en tu ubicación actual y permite seleccionarlos para ver la ruta hacia ellos.

Hospital Finder (ArogyaOnline) Localiza hospitales en un radio de 2, 4 y 8 kilómetros de distancia, da información detallada de cada hospital, tiene vista en modo mapa y proporciona indicaciones para llegar al hospital con la ruta calculada como más rápida así como tiempo estimado de llegada.

Hospital Finder (Darshan) proporciona información acerca de sí los hospitales son públicos o privados, proporciona el sitio web y correo electrónico del mismo.

Sin entrar en mucho detalle, la funcionalidad de estas aplicaciones dista mucho de su descripción en la PlayStore. La mayoría no proporciona una lista confiable de hospitales. Se especula que obtienen sus datos del mismo mapa y no un data set de hospitales registrados. Otros puntos en contra de las mismas son: la publicidad que en ocasiones aparece en cuanto se abre la app, el manejo de los datos y los permisos que solicitan, ninguna estima un tiempo de espera, muestran consultorios privados, nutriólogos y centros de salud no adecuados para emergencias.

Asimismo, encontramos en la literatura la importancia de la optimización de tiempos de espera ya que largos tiempos pueden reflejar la ineficiencia de las operaciones hospitalarias (Yaduvansh 2018) y los retrasos excesivos pueden ser perjudiciales en la salud de los pacientes (Patrick et al. 2008). Sin embargo, la medida de tiempo comienza desde que se ingresa al hospital hasta que se atiende, cuando un arribo más rápido a este podría optimizar el tiempo para brindar el servicio.

1. **Resultados (Hasta el momento)**

Hasta el momento tenemos un dataset el cual es una unión de 5 datasets diferentes ya que la información que necesitamos no se encontraba por si sola. El dataset contiene los recursos de salud de los hospitales en la ZMG (e.g. Número de médicos por hospital, camas disponibles, consultorios), Registros de urgencias, Egresos hospitalarios, coordenadas geográficas de los hospitales dentro de la ZMG Estos datos incluyen Hospitales y Clínicas. De igual manera contamos con

la API ya funcionando que calcula el tiempo de desplazamiento y distancia tomando en cuenta el tráfico de un punto geográfico a otro. Esperemos esta semana empezaremos a implementar el algoritmo de machine learning para hacer esta búsqueda de hospitales más eficiente y comenzar a relacionar las variables del dataset con tiempos de espera en los hospitales.

1. **Conclusiones finales y siguientes pasos**

CONSEGUIR MÁS DATOS. INCORPORAR LA API DE MAPAS. ATACAR EL CÓMO ESTIMAR EL TIEMPO DE ESPERA.

1. **Bibliografía**

* Secretaria de Salud (2018). Recursos Salud Nivel central. 2020, de Secretaria de Salud Sitio web: <https://datos.gob.mx/busca/dataset/recursos-en-salud-nivel-central>
* Secretaria de Salud (2015). Tasa de Médicos por mil habitantes . 2020, de Secretaria de Salud Sitio web: <https://datos.gob.mx/busca/dataset/indicadores-del-protocolo-de-san-salvador/resource/f2f69407-f5e2-4d9d-a3ab-fb13623d77e1>
* Secretaria de Salud (2018). Egresos Hospitalarios. 2020, de Secretaria de Salud Sitio web: <https://datos.gob.mx/busca/dataset/egresos-hospitalarios-de-la-secretaria-de-salud>
* Secretaria de Salud (2018). Urgencias Hospitalarias. 2020, de Secretaria de Salud Sitio web: <https://datos.gob.mx/busca/dataset/urgencias>
* INEGI (2019).Directorio Estadístico Nacional de Unidades Económicas. 2020, de INEGI. Sitio web: <https://www.inegi.org.mx/app/mapa/denue/>
* Yaduvanshi, Deepak. (2018). Application of queuing theory to optimize waiting time in hospital operations RCEM conference Feb 18.
* Patrick J, Puterman ML. Reducing Wait Times through Operations Research: Optimizing the Use of Surge Capacity. *Healthc Policy*. 2008;3(3):75‐88.

Notas:

Clinica del Imss asignada, solamente atenderte en ella?

KNN

Distancia Euclidiana, Mediante operación

Aprender con todos los elementos y depsues mediante el clustering.

