



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**FACULTAD DE CIENCIAS**

**TÍTULO DE LA TESIS**

**ELABORACIÓN DE UNA APP:  
PUMABUS NUEVA NORMALIDAD**

**QUE PARA OBTENER EL TÍTULO DE:**

**MATEMÁTICO**

**P R E S E N T A:**

**NOMBRE DEL ALUMNO**  
JOSE DANIEL AGUILERA MEJIA



FACULTAD DE CIENCIAS  
UNAM

**TUTORA**  
M. EN AUD. REYNA ELIZABETH  
CABALLERO CRUZ

**CIUDAD DE MÉXICO, 2022**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Índice

<b>Introducción.....</b>	<b>1</b>
Motivación .....	1
Objetivos generales .....	2
Estructura de la tesis.....	3
<b>Capítulo 1 .....</b>	<b>4</b>
<b>Marco teórico.....</b>	<b>4</b>
1.1.1 “El problema de la ruta más corta.....	4
1.1.2 Algoritmos para resolver rutas mínimas.....	5
1.1.3 Algoritmo Dijkstra.....	6
1.2 Población en Ciudad Universitaria.....	9
1.3 Estadísticas sobre el sistema de transporte interno Pumabús.....	14
1.3.1 Un poco de historia .....	14
1.3.2. Antecedentes.....	14
1.3.3 Datos estadísticos .....	16
1.4 Estadísticas sobre la pandemia en la Ciudad de México .....	18
<b>Capítulo 2 .....</b>	<b>22</b>
<b>Planeación de la aplicación.....</b>	<b>22</b>
2.1 ¿Como ayudará la aplicación? .....	22
2.1.1. Ventajas.....	22
2.1.2 Desventajas .....	23
2.2 Ubicación del Pumabús en tiempo real.....	23
2.2.1Modelo cliente/servidor .....	24
2.2.2 Ventajas.....	25
2.2.3 Desventajas .....	25
2.4 Parámetros para el cálculo del tiempo estimado. ....	25
<b>Capítulo 3 .....</b>	<b>27</b>
<b>Desarrollo de aplicación.....</b>	<b>27</b>
3.2 Desarrollo de la Aplicación .....	28
3.3 Algoritmo para la ruta más corta.....	42
3.3.1 Aplicación de algoritmo.....	42
<b>Capítulo 4 .....</b>	<b>45</b>
<b>Funcionamiento de la aplicación y resultados.....</b>	<b>45</b>
4.1 Funcionamiento de la aplicación y resultados. ....	45
4.1.1 Implementación en iOS.....	45

<b>Conclusiones</b> .....	<b>51</b>
<b>Bibliografía</b> .....	<b>53</b>
<b>Aapéndice A</b> .....	<b>55</b>



## Índice de Tablas

<b>Tabla 1.1 Estudiantes de Licenciatura por Facultad.....</b>	<b>10</b>
<b>Tabla 1.2 Estudiantes de nivel posgrado.....</b>	<b>11</b>
<b>Tabla 1. 3. Estudiantes por Facultad haciendo especialización.....</b>	<b>12</b>
<b>Tabla 1. 4 Población en Ciudad Universitaria.....</b>	<b>12</b>
<b>Tabla 1.5 Relación vueltas kilómetros (2013).....</b>	<b>16</b>
<b>Tabla 1.6 Porcentaje de usuarios por ruta.....</b>	<b>17</b>

## Índice de Figuras

<i>Figura 1.1 Contagios 27 abril 2020 .....</i>	<b>18</b>
<i>Figura 1. 1 Primer pico de contagios .....</i>	<b>11</b>
<i>Figura 1. 3 Segundo pico de contagios .....</i>	<b>20</b>
<i>Figura 1. 4 Contagios hasta mayo 2021 .....</i>	<b>21</b>
<i>Figura 2.1 Ubicación en tiempo real.....</i>	<b>23</b>
<i>Figura 3.2.1 Creación de proyecto.....</i>	<b>29</b>
<i>Figura 3.2.2 Selección de nombre de proyecto.....</i>	<b>30</b>
<i>Figura 3.2.3 Pantalla de inicio de sesión .....</i>	<b>31</b>
<i>Figura 3.3.1 Diagrama de la ruta más corta .....</i>	<b>43</b>
<i>Figura 4.1 Simulación en Iphone 11 .....</i>	<b>45</b>
<i>Figura 4.2 Pantalla SplashScreen.....</i>	<b>46</b>
<i>Figura 4.3 Pantalla principal.....</i>	<b>46</b>
<i>Figura 4.4 Menú Lateral .....</i>	<b>47</b>
<i>Figura 4.5 Ruta 4 .....</i>	<b>48</b>
<i>Figura 4.6 Mapa Ruta 4 .....</i>	<b>48</b>
<i>Figura 4.7 Botón de más información.....</i>	<b>49</b>
<i>Figura 4.8 Página más información.....</i>	<b>49</b>
<i>Figura 4.9 Botón Inicio de Sesión.....</i>	<b>50</b>
<i>Figura 4.10 Inicio de Sesión .....</i>	<b>50</b>

## Índice de códigos.

<b><i>Código 1.1 Algoritmo de Dijkstra.....</i></b>	<b>18</b>
<b><i>Código 3.2.1 MainPage.xaml .....</i></b>	<b>32</b>
<b><i>Código 3.2.2 Archivo MainPage.xaml .....</i></b>	<b>33</b>
<b><i>Código 3.2.3 Código de Ruta4.xaml .....</i></b>	<b>36</b>
<b><i>Código 3.2.4 Código de mapa estático .....</i></b>	<b>36</b>
<b><i>Código 3.2.5 Código para Google Maps .....</i></b>	<b>37</b>
<b><i>Código 3.2.6 Opciones de vista de mapa.cs.....</i></b>	<b>38</b>
<b><i>Código 3.2.7 Código de WebView.....</i></b>	<b>39</b>
<b><i>Código 3.2.8 Código de inicio de sesión .....</i></b>	<b>40</b>
<b><i>Código 3.2.9 Función barra de búsqueda.....</i></b>	<b>41</b>

## Introducción

El uso de las tecnologías en la lucha contra el covid19 ha modificado las formas de trabajar en todos los entornos, Ciudad Universitaria no es la excepción, el regreso a actividades presenciales esta cercano, se debe estar preparado para evitar contagios y asegurar la salud del todo el personal que forma parte de la Universidad Nacional Autónoma de México (UNAM).

Otro factor importante es tener una buena administración del tiempo para realizar las actividades dentro de Ciudad Universitaria.

## Motivación

Las aplicaciones móviles permiten generar un entorno de interacción único con los usuarios ya que cuenta con los objetivos claros y específicos; son utilizadas por las empresas para monitorear y gestionar procesos.

Por ello se plantea realizar una aplicación para teléfonos móviles que ayudará a tener un mejor control en el transporte dentro de la Universidad. El beneficio que se obtendrá es una menor propagación de la infección en términos globales, con la posibilidad de recuperar la libertad de acción, y una protección de la salud de los individuos.

Es importante mencionar que se han desarrollado trabajos de investigación donde se plantea la mejora del transporte en Ciudad Universitaria y la consecución de posicionamiento del PUMABÚS lo que es un monitoreo para la obtención de tiempo de llegada a alguna parada, que en parte este trabajo también lo plantea pero se trabaja sobre condiciones diferentes. Las principales diferencias con los otros trabajos es la obtención del cálculo del tiempo de llegada del autobús a alguna parada en tiempo real y el planteamiento de tener control de usuarios.

La importancia de implementar la aplicación es mayúsculo ya que un gran porcentaje de la población de la Universidad se beneficiaría de la misma, dado que al conocer el tiempo en el que llegará el autobús a la parada en la que lo esperan, podrán optimizar su tiempo para llegar a tiempo a la parada o tomar alguna

alternativa. El campus de la Universidad al ser tan grande y contar con una numerosa población se hace necesario implementar un sistema como el que se plantea en este trabajo, será de gran ayuda para quienes lo usen.

### Objetivos generales

- Proponer una aplicación para teléfonos móviles que brinde a los usuarios tiempos estimados de la llegada del Pumabús, a demás de ayudarles a moverse mejor dentro de Ciudad Universitria en el pumabus.
- Evitar conglomeraciones de gente en las paradas y por consiguiente contagios.
- Los usuarios podrán ver cada una de las paradas, al igual que dos mapas uno será interactivo y el otro será una imagen, con esto será muy difícil que se vuelvan a perder al tomar el Pumabús.
- Optimizar las rutas de los autobuses ya que no podrán ir al 100% de su capacidad actual, esto generará que mucha gente se quede esperando el autobús y puede ser un posible foco de infección por lo que con la aplicación se busca evitar esto, dando a conocer un tiempo estimado de llegada, también se busca mejorar la ruta con mayor demanda mediante el uso de la aplicación. Se quiere ver cuáles son las paradas con más usuarios por día y cual es la ruta con más usuarios por día para así poder generar una estrategia que ayude a que los autobuses hagan sus recorridos con la capacidad indicada y no dejar a mucha gente esperando en la parada.

## Estructura de la tesis

Para llevar a cabo todo lo mencionado la tesis se dividirá en 4 capítulos.

### **Capítulo 1**

Se darán a conocer estadísticas de la Universidad, como son población que hay dentro de la Universidad, mostrar estadísticas relevantes sobre el transporte de la Universidad Pumabús, datos sobre la pandemia en la Ciudad de México y como la aplicación pretende ayudar a reducir los contagios en el transporte Pumabús.

### **Capítulo 2**

En este capítulo se propone de como la aplicación va a apoyar la reducción el número de personas por parada se verán ventajas y desventajas del mismo. También se planteará el sistema para obtener la ubicación del Pumabús en tiempo real, por otra parte, se explicará el cálculo del tiempo estimado del autobús.

### **Capítulo 3**

En este capítulo se verá el diseño y desarrollo de la aplicación, también se verá a fondo el lenguaje con el que fue programada, se verán todos los recursos utilizados y características que Xamarin ofrece.

### **Capítulo 4**

En este capítulo se verá el funcionamiento de la aplicación, se harán simulaciones en ambos sistemas operativos, para conocer el funcionamiento de esta, de igual forma se verán los resultados de todo el proyecto.

Todos los datos recaudados son del ciclo escolar 2019-2020.

# Capítulo 1

## Marco teórico

En este capítulo se abordarán varios conceptos importantes para la realización del proyecto, además estadísticas de la Universidad, como son la población que hay dentro de la Universidad y porcentajes de usuarios por ruta del periodo de Enero de 2019 a Diciembre del 2020, además se muestran las estadísticas relevantes sobre el transporte de la Universidad Pumasabús, los datos sobre la pandemia por Covid-19 en la Ciudad de México

### 1.1.1 “El problema de la ruta más corta

*Este problema consiste en encontrar la ruta con menor longitud entre dos nodos específicos; es uno de los problemas más importantes de optimización combinatoria. A menudo se presenta en cierto tipo de actividades, en donde se requiere encontrar la ruta más corta entre dos nodos de una red, en la cual cada arco tiene un costo asociado, con el objetivo de minimizar el costo (costo, tiempo, longitud, etc.) total. Los algoritmos para este tipo de problemas han sido estudiados desde la época de los 50`s y continúan siendo área activa de investigación.*

*Los problemas de la ruta más corta se encuentran entre los más estudiados de la optimización de flujos de redes, cuentan interesantes aplicaciones en varios campos y resultan de gran interés, algunas de las razones son [1]*

- *Su amplia variedad de aplicaciones prácticas;*
- *La existencia de métodos de solución que al ser aplicados a una red con características específicas (sin ciclos negativos), proveen una solución exacta a un tiempo y costo razonables;*
- *La posibilidad de utilizarlo como inicio en el estudio de modelos complejos de redes;*
- *Su utilización como auxiliar en la búsqueda de soluciones a problemas para los que no existe un algoritmo exacto, mediante subrutinas.*

*El transporte es uno de los campos donde diferentes clases del problema de la ruta más corta requieren solución. Por la naturaleza de la aplicación, se necesitan procedimientos de la ruta más corta que sean muy flexibles y eficientes. Actualmente no existe el mejor algoritmo para todas las clases de problemas de la ruta más corta, la investigación en este campo se ha orientado al diseño e implementación de rutinas que sean “ad hoc” y consideren las peculiaridades del problema en cuestión (Lozano 2011).*

*Es posible encontrar el problema de la ruta más corta de tres formas distintas:*

- Del nodo fuente  $s$  al nodo sumidero  $t$ . En este caso debe existir al menos una ruta entre  $s$  y  $t$ .*
- Del nodo fuente  $s$  al nodo de la red  $i$ . Para este caso deben existir rutas de  $s$  a  $i$ . Este caso es conocido como el problema de árbol de rutas más cortas.*
- Entre todo par de nodos. Y para esta última forma, debe existir al menos, una ruta entre todo par de nodos.*

*En los tres casos mencionados, para que exista solución no pueden existir circuitos negativos en la red, es decir, circuitos que al tener arcos con costos negativos reduzcan el costo de la ruta cada vez que son recorridos.*

### *1.1.2 Algoritmos para resolver rutas mínimas*

*El problema de la ruta más corta y el SPT (short path tree) pueden resolverse utilizando programación lineal [2]. Planteándose como el envío de una unidad de flujo  $f$ , del nodo origen  $s$  al nodo destino  $t$  o al resto de los nodos de la red, según sea el caso, al mínimo costo. Esto es,  $f_s = 1$ ,  $f_t = -1$ , y  $f_i = 0$ , para  $i$  distinto de  $s$  ó  $t$ . Entonces, el planteamiento es como sigue:*

$$\text{Minimizar } \sum_{i=1}^t \sum_{j=1}^t c_{ij} x_{ij}$$

*Sujeto a:*



$$\sum_{j=1}^t x_{ij} - \sum_{k=1}^t x_{ki} = \begin{cases} 1 & \text{si } i = s \\ 0 & \text{si } i \neq s \text{ ó } t \\ -1 & \text{si } i = t \end{cases}$$

$$x_{ij} = 0 \text{ ó } 1 \quad i, j = 1, 2, \dots, t$$

Las ecuaciones de conservación de flujo son unimodulares, es decir, si existe una solución el método simplex obtendrá valores 1, 0. Debido a esta característica de las ecuaciones de conservación de flujo es posible replantear la última restricción como:

$$x_{ij} \geq 0 \quad i, j = 1, 2, \dots, t$$

Para el segundo caso se hace la modificación  $f_i = 0$  por  $f_i = -1$ .

Existen también otros algoritmos para la solución de estos problemas, por ejemplo, Dijkstra y Floyd. El primero permite obtener la solución para el SPT, es decir, la ruta más corta del nodo de origen al nodo destino o del nodo origen a todos los demás nodos de la red. Mientras que el algoritmo de Floyd proporciona la solución para la ruta más corta entre cualquier par de nodos de la red [1]. En este caso vamos a usar el algoritmo de Dijkstra que se explicará a fondo a continuación.

### 1.1.3 Algoritmo Dijkstra

Este algoritmo fue publicado por primera vez por Edsger Dijkstra en 1959, con la finalidad de solucionar el problema de la ruta más corta desde el nodo fuente "s" a todos los demás nodos de la red "i", siempre y cuando las distancias de los arcos sean no negativas.

Es considerado el algoritmo más eficiente para solucionar este tipo de problemas. Se basa en la asignación temporal de etiquetas a los nodos. La etiqueta inicial es una cota superior de la longitud de la ruta del nodo s al nodo i.

Como el proceso es iterativo las etiquetas se reducen en cada iteración (etiquetas temporales), hasta obtener la mínima longitud de la ruta de  $s$  a  $i$  (etiquetas permanentes).

Sea  $G = (V, E, d)$  un grafo dirigido y con distancias no negativas en sus arcos, y sea  $s$  el nodo fuente (raíz) de  $G$ . Para cada nodo  $v$  en  $G$  se busca encontrar la ruta desde  $s$  a  $v$ ,  $P(s,v)$ , tal que la suma de las distancias de los arcos en  $P(s,v)$  sea mínima. El algoritmo de Dijkstra obtiene la arborescencia de rutas más cortas del nodo fuente  $s$  a todo nodo de la red  $i$ . El procedimiento consiste en elegir el arco para agregar en cada paso considerado que es el único que minimiza  $d(s,k) + longitud(k,i)$ . La Figura 3.19 se presentan los pasos del algoritmo de Dijkstra Algoritmo Dijkstra [3]

Datos de entrada: un grafo dirigido  $G = (V, E, d)$ , donde cada arco  $(i,j)$  tiene una longitud no negativa,  $d(i,j)$ .

**Paso 1** indicación de etiquetas

Etiqueta permanente:  $d(s) = 0$

Etiquetas temporales:  $d(i) = \infty$  para toda  $i \neq s$

$a(i) = s$  donde  $s$  es el predecesor de  $i$

$p = s$  el nodo  $s$  tiene etiqueta permanente

**Paso 2** Actualización de etiquetas

Sea  $E^+$  el conjunto de arcos con trayectoria positiva

Para toda  $i \in E^+(p)$  con etiqueta temporal, actualizar de la siguiente forma:

$$d(i) = \min\{d(i), d(p) + d(p,i)\}$$

Si  $d(i)$  se modificó, entonces  $a(i) = p$

Sea  $i^*$  tal que  $d(i^*) = \min\{d(i) \mid d(i) \text{ es temporal}\}$

Si  $d(i^*) = \infty$  terminar, no existe arborescencia de raíz  $s$

En otro caso  $d(i^*)$  es etiqueta permanente y hace  $p = i^*$

### Paso 3

Si todos los nodos tienen etiqueta permanente, terminar.

En otro caso, ir al paso 2.

Nota: si la ruta es de  $s$  a  $k$ , entonces el algoritmo se modifica en el paso 3 como sigue:

Si  $p = k$  terminar, y la distancia del nodo  $s$  al nodo  $i$  es  $d(i)$ .

Si  $p \neq k$  ir al paso 2"

Fuente: Soto, M. (2013) Modelación de hiper rutas mínimas multimodales del transporte público de pasajeros: zona centro sur del DF. Maestro en ingeniería. Universidad Nacional Autónoma de México.

---

Algorithm 1: Dijkstra ()

---

```
1 begin
2    $S := \phi : S' := N;$ 
3   foreach  $i \in N$  do
4      $d(i) := \infty;$ 
5   end
6    $d(s) := 0$  and  $pred(s) := 0;$ 
7   while  $|S| < n$  do
8     begin
9       let  $i \in S'$  be a node for which  $d(i) = \min\{d(j) : j \in S'\};$ 
10       $S = S \cup \{i\};$ 
11       $S' = S' - \{i\};$ 
12      foreach  $(i, j) \in A(i)$  do
13        if  $d(j) > d(i) + C_{ij}$  then
14           $d(j) := d(i) + C_{ij}$  and  $pred(j) := i;$ 
15        end
16      end
17    end
18  end
19 end
```

---

Código 1.1 Algoritmo de Dijkstra Fuente: Sung et al. (2000)

## 1.2 Población en Ciudad Universitaria

Considerando que la población total se compone de estudiantes, docentes, administrativos y visitantes externos, es importante conocer el tamaño de la población para poder realizar un sistema de cómputo que cumpla su propósito. Para ello el dato conocido es el número total de estudiantes, docentes y administrativos.

Primero se mostrarán los datos relacionados con estudiantes por Facultad a nivel licenciatura, después cuantos hay a nivel posgrado, cuántos hay a nivel especialización y por último el número de docentes y de la planta administrativa.

Para realizar se consultó la Agenda Estadística UNAM 2020 de la Dirección General de Planeación. En la tabla 1.1 se muestra el número de estudiantes por Facultad a nivel licenciatura.

El total de estudiantes de todas las Facultades es: 118,758, excluyendo a la Facultad de Música, ya que no se encuentra dentro de Ciudad Universitaria.

Entidad académica	Primer ingreso			Reingreso			Población total
	Hombres	Mujeres	Total	Hombres	Mujeres	Total	
Facultad de Arquitectura	695	586	1,281	3,064	3,029	6,093	7,374
Facultad de Artes y Diseño	229	535	764	749	1,695	2,444	3,208
Facultad de Ciencias	1,204	919	2,123	4,129	3,213	7,342	9,465
Facultad de Ciencias Políticas y Sociales <sup>a</sup>	1,569	1,785	3,354	4,015	4,852	8,867	12,221
Facultad de Contaduría y Administración <sup>a</sup>	1,978	2,107	4,085	6,552	5,530	12,082	16,167
Facultad de Derecho <sup>a</sup>	1,666	2,032	3,698	4,976	5,852	10,828	14,526
Facultad de Economía <sup>a</sup>	901	640	1,541	3,408	1,640	5,048	6,589
Facultad de Filosofía y Letras <sup>a</sup>	1,156	1,594	2,750	3,592	5,391	8,983	11,733
Facultad de Ingeniería	1,898	642	2,540	7,521	2,458	9,979	12,519
Facultad de Medicina	556	1,119	1,675	2,041	3,906	5,947	7,622
Facultad de Medicina Veterinaria y Zootecnia	149	453	602	796	1,898	2,694	3,296
Facultad de Música	98	55	153	333	196	529	682
Facultad de Odontología	130	416	546	613	1,615	2,228	2,774
Facultad de Psicología <sup>a</sup>	261	637	898	828	2,183	3,011	3,909
Facultad de Química	657	808	1,465	2,755	3,135	5,890	7,355

Tabla 1.1 Estudiantes de Licenciatura por Facultad. Fuente: UNAM(2020) Agenda Estadística UNAM 2020. Cuadernos de Planeación Universitaria Año 2020. <https://www.planeacion.unam.mx/Agenda/2020/pdf/Agenda2020.pdf>

En la Tabla 1.1 apreciamos los estudiantes que hay por cada Facultad dentro de Ciudad Universitaria, tanto los que son de primer ingreso como los que son de reingreso, también nos muestra cuántos de estos son hombres y cuántos son mujeres.

La Tabla 1.2 muestra el número total de estudiantes a nivel posgrado, como se observa son 4 principales áreas: Ciencias físico matemáticas e ingenierías, Ciencias biológicas, químicas y de la salud, Ciencias sociales y Humanidades y artes. Si se suman todos los estudiantes que hay estudiando algún posgrado nos da un total de 13,636 que se sumarán al total de estudiantes por Facultad de 118,758 mencionado anteriormente.

Área / Programa	Primer ingreso			Reingreso			Población total
	Hombres	Mujeres	Total	Hombres	Mujeres	Total	
<b>Ciencias físico matemáticas e ingenierías</b>	<b>734</b>	<b>273</b>	<b>1,007</b>	<b>1,290</b>	<b>489</b>	<b>1,779</b>	<b>2,786</b>
Maestría en Docencia para la Educación Media Superior	11	12	23	14	18	32	55
Posgrado en Astrofísica	34	15	49	48	27	75	124
Posgrado en Ingeniería	385	146	531	592	216	808	1,339
Posgrado en Ciencia e Ingeniería de la Computación	57	12	69	63	19	82	151
Posgrado en Ciencia e Ingeniería de Materiales	39	27	66	74	56	130	196
Posgrado en Ciencias de la Tierra	38	25	63	140	94	234	297
Posgrado en Ciencias Físicas	74	15	89	139	16	155	244
Posgrado en Ciencias Matemáticas <sup>a</sup>	96	21	117	220	43	263	380
<b>Ciencias biológicas, químicas y de la salud</b>	<b>589</b>	<b>797</b>	<b>1,386</b>	<b>1,498</b>	<b>1,636</b>	<b>3,134</b>	<b>4,520</b>
Doctorado en Ciencias Biomédicas	35	41	76	175	203	378	454
Maestría en Ciencias Neurobiología	20	19	39	22	16	38	77
Maestría en Enfermería	4	10	14	9	33	42	56
Maestría en Docencia para la Educación Media Superior	8	19	27	20	39	59	86
Maestría y Doctorado en Ciencias Bioquímicas	100	111	211	232	175	407	618
Maestría y Doctorado en Ciencias de la Producción y de la Salud Animal	54	80	134	114	152	266	400
Maestría y Doctorado en Ciencias Médicas, Odontológicas y de la Salud	70	123	193	196	261	457	650
Maestría y Doctorado en Ciencias Químicas	68	33	101	214	121	335	436
Maestría y Doctorado en Psicología	58	139	197	137	223	360	557
Posgrado en Ciencias Biológicas	113	137	250	250	246	496	746
Posgrado en Ciencias del Mar y Limnología	34	41	75	60	77	137	212
Posgrado en Ciencias de la Sostenibilidad	18	38	56	24	73	97	153
Programa combinado en Medicina (Licenciatura y Doctorado)	7	6	13	45	17	62	75
<b>Ciencias sociales</b>	<b>801</b>	<b>767</b>	<b>1,568</b>	<b>1,383</b>	<b>1,298</b>	<b>2,681</b>	<b>4,249</b>
Maestría en Trabajo Social	6	21	27	14	42	56	83
Maestría en Docencia para la Educación Media Superior	14	27	41	12	27	39	80
Posgrado en Antropología	11	19	30	23	34	57	87
Posgrado en Ciencias de la Administración	404	415	819	719	665	1,384	2,203
Posgrado en Ciencias Políticas y Sociales	73	66	139	134	120	254	393
Posgrado en Derecho	183	155	338	293	272	565	903
Posgrado en Economía	63	19	82	87	48	135	217
Posgrado en Estudios Latinoamericanos	30	24	54	62	56	118	172
Posgrado en Geografía	17	21	38	39	34	73	111
<b>Humanidades y artes</b>	<b>353</b>	<b>346</b>	<b>699</b>	<b>697</b>	<b>685</b>	<b>1,382</b>	<b>2,081</b>
Maestría en Diseño Industrial	5	4	9	6	7	13	22
Maestría en Docencia para la Educación Media Superior	16	31	47	26	34	60	107
Maestría y Doctorado en Arquitectura	40	42	82	74	53	127	209
Maestría y Doctorado en Bibliotecología y Estudios de la Información <sup>a</sup>	11	14	25	29	55	84	109
Maestría y Doctorado en Estudios Mesoamericanos	9	6	15	32	36	68	83
Maestría y Doctorado en Filosofía	23	12	35	70	32	102	137
Maestría y Doctorado en Filosofía de la Ciencia	34	18	52	51	39	90	142
Maestría y Doctorado en Historia	34	19	53	38	30	68	121
Maestría, Doctorado y Especialización en Historia del Arte	21	37	58	55	76	131	189
Maestría y Doctorado en Letras	5	3	8	44	39	83	91
Maestría y Doctorado en Lingüística	15	12	27	19	21	40	67
Maestría y Doctorado en Música <sup>a</sup>	32	12	44	38	21	59	103
Maestría y Doctorado en Pedagogía	36	55	91	81	130	211	302
Maestría y Doctorado en Urbanismo	17	21	38	45	42	87	125
Posgrado en Artes y Diseño	55	60	115	89	70	159	274

Tabla 1.2 Estudiantes de nivel posgrado Fuente: UNAM(2020) Agenda Estadística UNAM 2020. Cuadernos de Planeación Universitaria Año 2020. <https://www.planeacion.unam.mx/Agenda/2020/pdf/Agenda2020.pdf>

En la Tabla 1.3 se muestra el número de estudiantes que cursan alguna especialización por Facultad.

Entidad académica	Primer ingreso			Reingreso			Población total
	Hombres	Mujeres	Total	Hombres	Mujeres	Total	
Facultad de Arquitectura	106	103	209	71	93	164	373
Facultad de Ciencias	3	2	5	0	0	0	5
Facultad de Ciencias Políticas y Sociales	40	31	71	2	6	8	79
Facultad de Contaduría y Administración	110	160	270	102	135	237	507
Facultad de Derecho	424	484	908	282	306	588	1,496
Facultad de Economía	95	58	153	8	5	13	166
Facultad de Filosofía y Letras	11	16	27	0	0	0	27
Facultad de Ingeniería	161	47	208	86	33	119	327
Facultad de Medicina	1,870	2,142	4,012	3,855	4,085	7,940	11,952
Facultad de Medicina Veterinaria y Zootecnia <sup>a</sup>	64	46	110	12	2	14	124
Facultad de Odontología	54	98	152	39	116	155	307
Facultad de Psicología	7	8	15	6	42	48	63
Facultad de Química	2	3	5	2	4	6	11

Tabla 1. 3. Estudiantes por Facultad cursando alguna especialización. Fuente: UNAM(2020) Agenda Estadística UNAM 2020. Cuadernos de Planeación Universitaria Año 2020.  
<https://www.planeacion.unam.mx/Agenda/2020/pdf/Agenda2020.pdf>

El total de estudiantes que cursan una especialización es de 15,437.

Para el personal académico tenemos un total de 20322.

La tabla 1.4 muestra la población total en Ciudad Universitaria.

<b>Estudiantes por Facultad</b>	<b>118,758</b>
<b>Estudiantes de Posgrado</b>	<b>13,636</b>
<b>Estudiantes por Especialidad</b>	<b>20,322</b>
<b>Total Estudiantes</b>	<b>168,153</b>

Tabla 1. 4 Población en Ciudad Universitaria

En la Tabla 1.4 se observa que el grueso de la población se encuentra en el segmento de estudiantes por Facultad. Este será incluido en el desarrollo de la aplicación.

Con base en estos datos se realizará un análisis del total de personas que tiene cada Facultad y parada por Facultad con la finalidad de evitar conglomeraciones.



## 1.3 Estadísticas sobre el sistema de transporte interno Pumabús

### 1.3.1 Un poco de historia

*“El servicio de transporte interno en Ciudad Universitaria desde sus inicios ha venido modificándose conforme al paso de los años. En la década de los noventa se contaba con solo tres rutas, de las que se derivaban otras dos, una para la Zona Cultural y otra para el Jardín Botánico.*

*El parque vehicular con el que se ha prestado el servicio de transporte por parte de la Dirección General de Servicios Generales de la UNAM ha sido diverso, desde de camiones, microbuses y hasta camiones que pertenecieron a la Ruta 100 después de la desaparición de ésta.*

*En los últimos años, con el incremento del uso del automóvil en Ciudad universitaria y al estacionar éstos en los circuitos donde se encuentran las Facultades, se producía un tráfico intenso contándose con un solo carril para circular. Por lo anterior, se tuvo la necesidad de ya no dejar estacionar los vehículos en los principales circuitos para así liberar los carriles ocupados y así asignar un carril exclusivo para el transporte interno. A cambio los automovilistas tienen como alternativa estacionar sus vehículos en los estacionamientos del Estadio Olímpico Universitario”*

Fuente: Wikipedia. (2020). De <https://es.wikipedia.org/wiki/Pumab%C3%BAs>

### 1.3.2. Antecedentes

*“Antes de 2007, el transporte interno en Ciudad Universitaria contaba con:*

- *38 unidades.*
- *6 rutas, las cuales sumaban 58.6 kilómetros en sus recorridos, cubriendo un aproximado de 65.47% de las vialidades del campus CU.*
- *73 paraderos.*
- *En ese entonces, se permitía que los vehículos se estacionaran en los carriles laterales de las vialidades, por lo que las unidades transitaban en un*

*solo carril con el resto de los vehículos, lo que provocaba un intenso tráfico al interior del campus, así como contaminación atmosférica, sonora y visual.*

*A partir de 2007, se hacía necesario modernizar el transporte interno y para ello, se concibe como un sistema integral, surge entonces el concepto Pumabús, lo que implicó:*

- *El confinamiento de un carril exclusivo.*
- *El retiro de vehículos de las vialidades del circuito escolar.*
- *La habilitación de 7 estacionamientos remotos en el estadio olímpico para el reacomodo de los vehículos en 2,400 cajones.*

*A partir del año 2008 el sistema se fue consolidando, con la ampliación del servicio, al circuito exterior del campus CU, destacando las siguientes acciones:*

- *Mejora de infraestructura.*
- *Adquisición de nuevas unidades.*
- *Ampliación de 6 a 12 rutas, sumando 61.2 km en sus recorridos y cubriendo aproximadamente el 65.47% de las vialidades del campus.*
- *Capacitación de operadores .*
- *Sesibilización de la comunidad universitaria, a través de pláticas y campañas de difusión en materia de educación vial*
- *Integración de “Brigadas viales”*

Fuente: Secretaría de Servicios a la Comunidad. (2014). De <http://dicyg.fi-c.unam.mx:8080/sistemas/PUMABUSUNAM.pdf>

### 1.3.3 Datos estadísticos

“En la actualidad, el sistema cuenta con 65 unidades, 96 paraderos y traslada a 146,000 usuarios al día. Las tablas 1.5 y 1.6 muestran el número de vueltas y el porcentaje de usuarios por ruta respectivamente”

<i>Mes</i>	<i>Kilómetros recorridos</i>	<i>Numero de Vueltas</i>
<i>Enero</i>	179,481	24,928
<i>Febrero</i>	196,992	27,360
<i>Marzo</i>	141,696	19,680
<i>Abril</i>	248,371	34,496
<i>Mayo</i>	232,243	32,256
<i>Junio</i>	198,144	27,520
<i>Julio</i>	66,355	9,216
<i>Agosto</i>	197,704	27,456
<i>Septiembre</i>	184,320	25,600
<i>Octubre</i>	217,267	30,176
<i>Noviembre</i>	175,104	24,320
<i>Diciembre</i>	80,870	11,232

Tabla 1.5 Relación vueltas kilómetros. Fuente: Barrientos, C. Hernández, I. (2015, p. 15, Tabla 1.1 Relación Kilómetros Vueltas de 2013) Sistema de Geoposicionamiento del Pumabús, Licenciatura de Ingeniero en Computación. Universidad Nacional Autónoma de México.

<i>Ruta</i>	<i>Usuarios por día</i>	<i>Paraderos</i>	<i>Longitud de ruta</i>
<i>1</i>	16.50%	16	7.2
<i>2</i>	17.00%	11	4.2
<i>3</i>	9.00%	20	8.8
<i>4</i>	3.50%	15	9.7
<i>5</i>	13.00%	18	8.3
<i>6</i>	3.50%	24	10.2
<i>7</i>	11.00%	17	4.8
<i>8</i>	10.50%	20	5.8
<i>9</i>	14.00%	15	4.1
<i>10</i>	1.30%	18	11
<i>11</i>	0.30%	14	5.8
<i>12</i>	0.04%	7—8	5.1

*Tabla 1.6 Porcentaje de usuarios por ruta. Fuente: Barrientos, C. Hernández, I. (2015, p. 15, Tabla 1.2 Tabla de Usuarios 2013) Sistema de Geoposicionamiento del Pumabús, Licenciatura de Ingeniero en Computación. Universidad Nacional Autónoma de México.*

Fuente: Barrientos, C. Hernández, I. (2015) Sistema de Geoposicionamiento del Pumabús, Licenciatura de Ingeniero en Computación. Universidad Nacional Autónoma de México.

Cabe mencionar que se solicitaron los datos a las oficinas de la Dirección General de Servicios Generales y Movilidad (DGSGM) en varias ocasiones, pero no se tuvo respuesta satisfactoria a las peticiones. Por lo tanto, se realizó una investigación documental.

Cuando se regresen a clases presenciales los autobuses no podrán ocupar el 100% de su capacidad ocuparán un porcentaje que se desconoce por lo que será necesario implementar una estrategia para poder satisfacer las necesidades de los usuarios, la aplicación propondrá una posible situación a dicho problema.

#### 1.4 Estadísticas sobre la pandemia en la Ciudad de México

Es importante abordar las estadísticas del número de contagios de la pandemia Covid-19 en la Ciudad de México porque existe un alto riesgo de contagio en las estaciones de transporte del PUMABÚS dentro de la Ciudad Universitaria por lo que se pretende incluir esta información para determinar el nivel de utilidad de la aplicación, según la tendencia actual de dichas estadísticas.

El primer caso de COVID-19 en México se presentó el 27 de febrero del 2020, a partir de esa fecha la pandemia creció exponencialmente como se observa en la Figura 1.1.

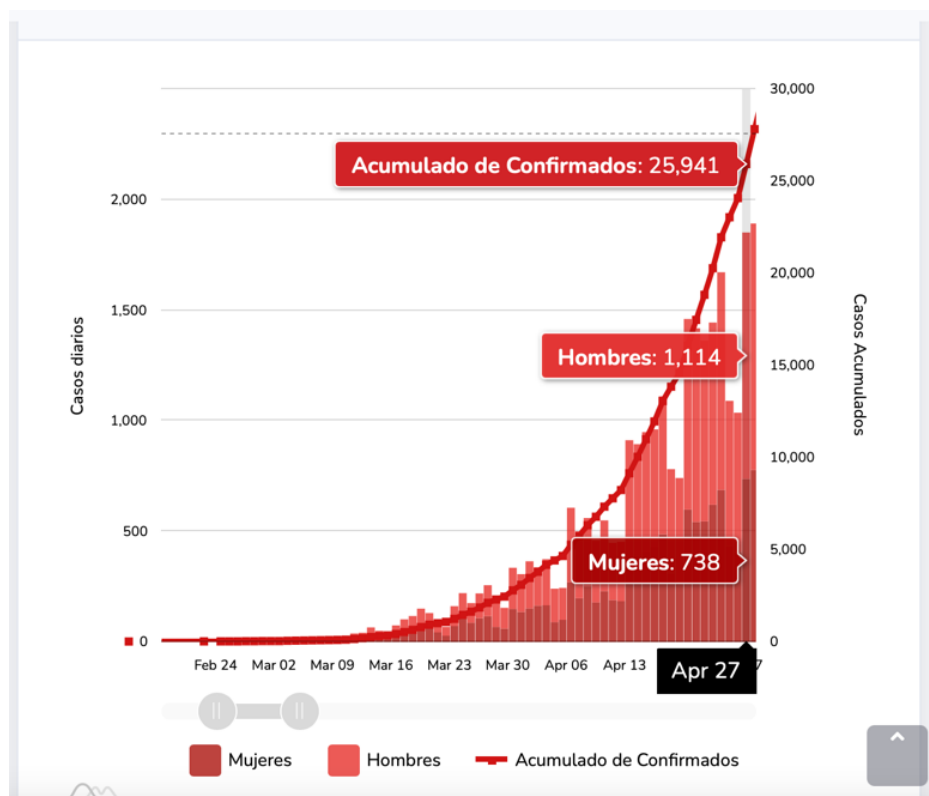


Figura 1.1 Contagios 27 abril 2020 Fuente: Gobierno de México Covid-19 México. (2020). De <https://datos.covid-19.conacyt.mx>

La figura 1.1 nos muestra el número total de contagios que hubo a tan solo 2 meses de darse el primer caso en el país, para el 27 de abril del 2020 ya se contaba con más de 25 mil infectados.

El primer pico (número máximo de contagios) de la pandemia se presentó en el mes de septiembre del 2020. Se muestra en la Figura 1.2.

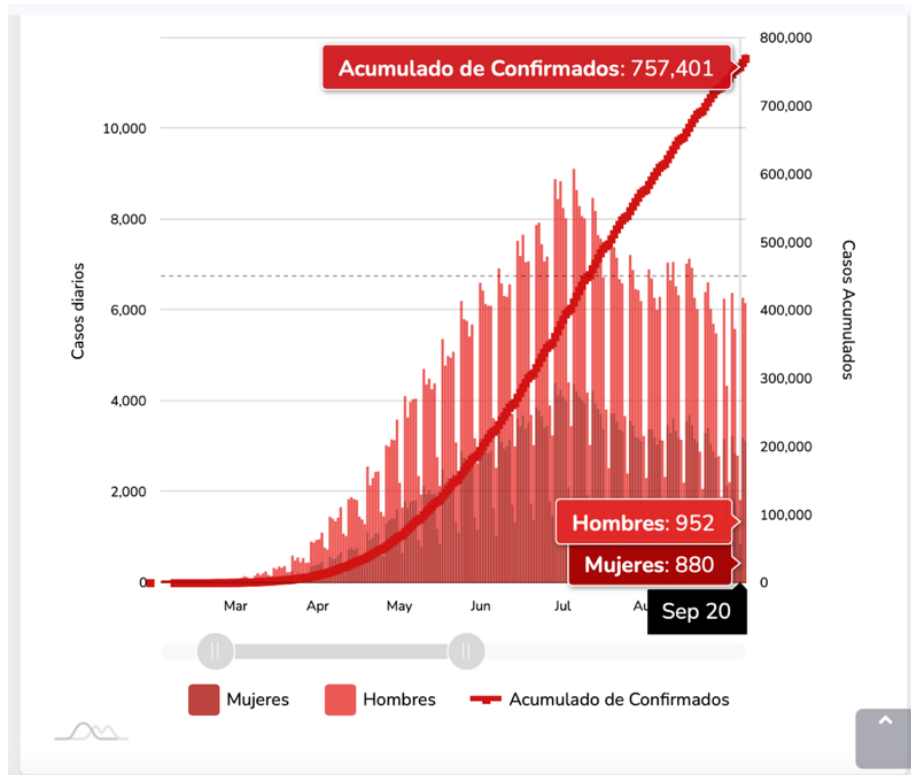


Figura 1. 2 Primer pico de contagios. Contagios 20 Septiembre 2020 Fuente: Gobierno de México Covid-19 México. (2020). De <https://datos.covid-19.conacyt.mx>

En la Figura 1.2 muestra una cantidad de contagios que van de 0 a 100,000 casos acumulados y de forma exponencial subsecuentemente de abril a julio, mientras que a finales de agosto se muestra un decremento paulatino y hasta septiembre.

A partir de septiembre de 2020 se mantendría creciente la curva de contagios durante los próximos 2 meses hasta diciembre, mes en el que vendría el segundo pico. Por lo tanto es importante haber mencionado el escenario de contagios en el país porque este mismo comportamiento se repitió cinco veces más.

Dado que este escenario se siguió presentando como se mencionó, el análisis para los siguientes picos de contagios es similar y el objeto de estudio en este trabajo justifica el desarrollo de la aplicación.

La Figura 1.3 muestra el segundo pico de contagios de la pandemia.

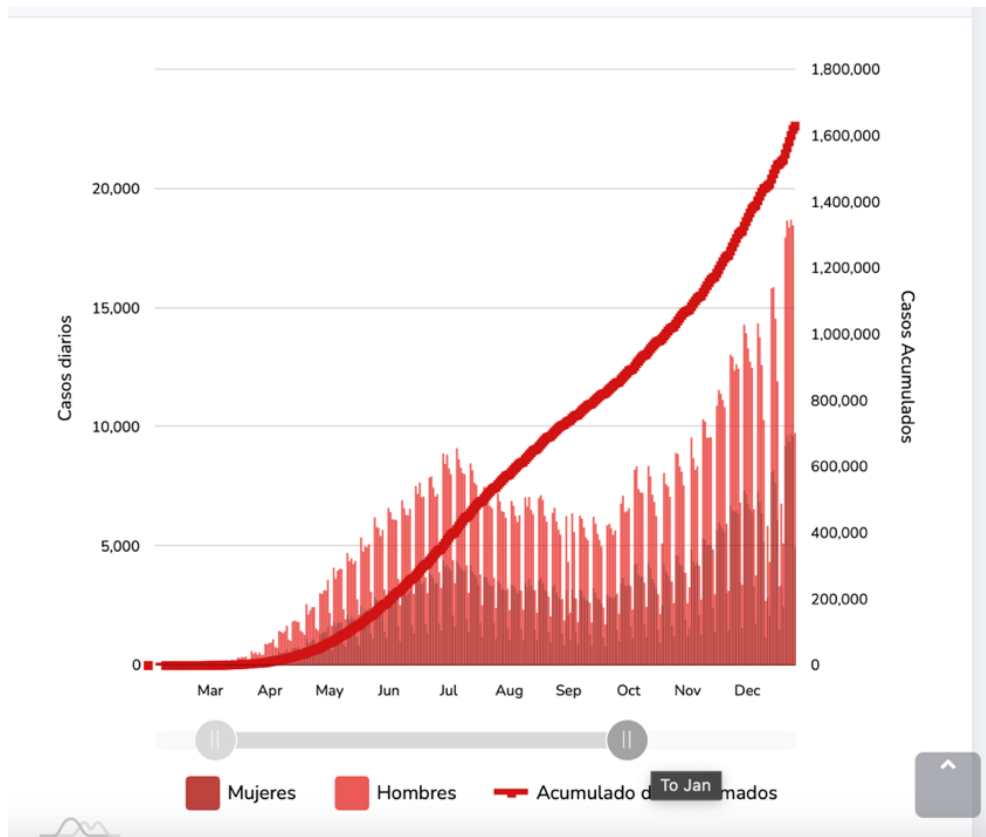


Figura 1. 3 Segundo pico de contagios. Contagios 15 de Diciembre de 2020 Fuente: Gobierno de México Covid-19 México. (2020). De <https://datos.covid-19.conacyt.mx>

De enero hasta el 24 de mayo del 2021 la curva de contagios fue disminuyendo.

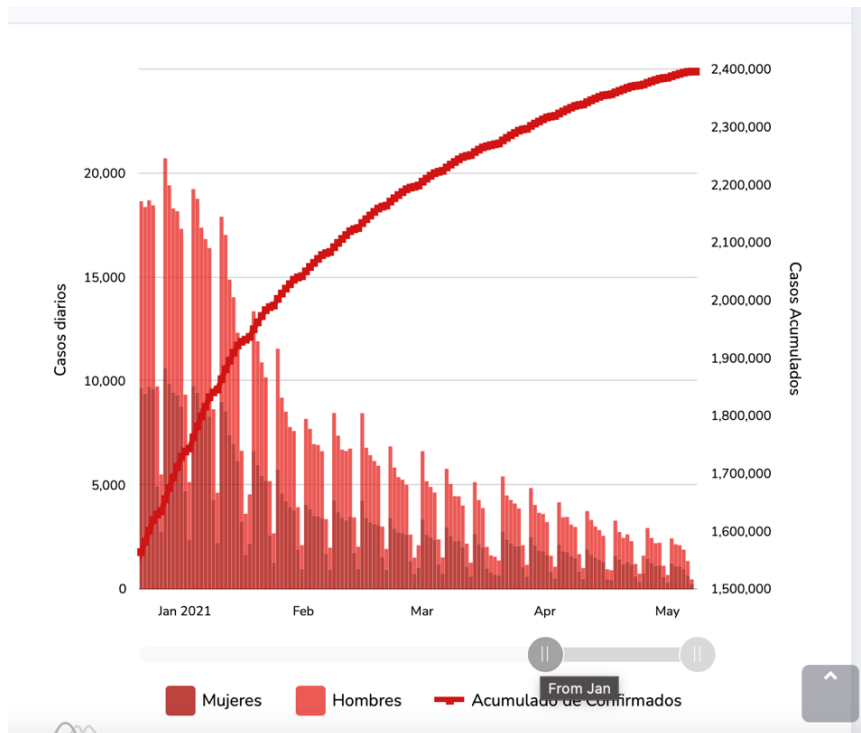


Figura 1. 4 Contagios hasta mayo 2021. Contagios 24 de Mayo de 2021 Fuente: Gobierno de México Covid-19 México. (2021). De <https://datos.covid-19.conacyt.mx>

Para el día 09 de agosto del 2022 se obtuvo la cifra de 6 millones 875 mil 708 contagios y 328 mil 437 defunciones en todo el país, no hay fecha exacta sobre cuando terminarán los contagios en el país, debemos considerar que al regresar a clases presenciales todos corremos el riesgo de contagiarnos a pesar de todas las medidas de seguridad que se van a tomar.

Es importante conocer los datos de la pandemia para tener presente que todos estamos en riesgo constante de infectarnos, por lo que no hay que ceder ante las indicaciones que realiza la Universidad, llevar a cabo todas las medidas sanitarias, siempre tener en cuenta la sana distancia y lo más importante seguirnos cuidando.



## Capítulo 2

### Planeación de la aplicación

En este capítulo se propone la estrategia de como la aplicación apoyará la reducción del número de personas por parada, se verán ventajas y desventajas del mismo. También se expone el sistema de cómputo para obtener la ubicación del Pumabús en tiempo real; por último se explicará el cálculo del tiempo estimado del autobús.

Es de gran importancia evitar conglomeraciones en las paradas dadas las circunstancias en las que nos encontramos, hay muchos contagios en la Ciudad y en CU, por lo que se busca evitar que tanto los alumnos como el personal académico y administrativo se contagien.

#### 2.1 ¿Como ayudará la aplicación?

Se busca que las personas que pertenecen a la comunidad de la UNAM se registren o hagan un inicio de sesión en la aplicación. Con lo anterior se plantea tener un control acerca del número de usuarios que se encuentren usando la aplicación y de esta forma conocer en tiempo real cuál es la parada con más gente para así proponer una solución a este problema.

Además, se mostrará en tiempo real el tiempo que tardará en llegar el autobús a cara parada y los usuarios tendrán conocimiento del mismo, por lo que en vez de esperar 10 ó 15 minutos el autobús, el usuario podrá calcular el tiempo para llegar a la parada y tomar el mismo, evitando la conglomeración de personas.

##### 2.1.1. Ventajas

Se tiene conocimiento del tiempo real en el que llegará el autobús a la parada, se posee control acerca del número de usuarios están usando la aplicación y cuales paradas son las más demandadas. También podrá ayudar a los usuarios ya que contará con una búsqueda de las paradas conocidas, así como las rutas en las que los usuarios podrán ingresar su punto de origen y su destino; la aplicación mediante un algoritmo les mostrará la ruta más corta o bien pueden ser canales directos que

conecten con el usuario para notificar o realizar gestiones como es el caso de las instituciones públicas.

### 2.1.2 Desventajas

Las aplicaciones consumen espacio, tiempo y en su mayoría datos, debido a que requieren de conexión a Internet. Es importante la experiencia del usuario en cuanto a (navegación, interacción, funcionamiento, entre otros).

### 2.2 Ubicación del Pumabús en tiempo real

El sistema está diseñado para utilizarse en dispositivos móviles y/o tabletas de los usuarios, como un dispositivo receptor y emisor. Su funcionamiento es de la siguiente forma:

1. El usuario al estar en la parada y hacer uso de la aplicación y otro usuario que vaya en un autobús de igual forma que tenga la aplicación abierta de esta forma ambos dispositivos se convierten en emisores, mandan la información a un servidor que recibe la información y la actualiza.
2. Para un mejor resultado se puede hacer que el conductor de cada autobús tenga la aplicación abierta mientras conduce, esto ayuda a que el 100% de los autobuses estén monitoreados.



Figura 2. 1 Ubicación en tiempo real

Para analizar la información, los servidores la recibirán y mostrarán en la aplicación de los usuarios y conductores. Se basa en el modelo cliente/servidor.

### 2.2.1 Modelo cliente/servidor

*“La arquitectura cliente servidor tiene dos partes claramente diferenciadas, por un lado la parte del servidor y por otro la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software específico que actúa de depósito de datos y funcione como un sistema gestor de base de datos o aplicaciones.*

#### Componentes

*Para entender este modelo vamos a nombrar y definir a continuación algunos conceptos básicos que lo conforman.*

- *Red: Una red es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.*
- *Cliente: El concepto de cliente hace referencia a un demandante de servicios, este cliente puede ser un ordenador como también una aplicación de informática, la cual requiere información proveniente de la red para funcionar.*
- *Servidor: Un servidor hace referencia a un proveedor de servicios, este servidor a su vez puede ser un ordenador o una aplicación informática la cual envía información a los demás agentes de la red.*
- *Protocolo: Un protocolo es un conjunto de normas o reglas y pasos establecidos de manera clara y concreta sobre el flujo de información en una red estructurada.*
- *Servicios: Un servicio es un conjunto de información que busca responder las necesidades de un cliente, donde esta información pueden ser mail, música, mensajes simples entre software, videos, etc.*

- *Base de datos: Son bancos de información ordenada, categorizada y clasificada que forman parte de la red, que son sitios de almacenaje para la utilización de los servidores y también directamente de los clientes”*

Fuente: Schiaffarino A. Modelo Cliente Servidor. Infranetworking. <https://normas-apa.org/referencias/citar-pagina-web/>

### 2.2.2 Ventajas

La gran ventaja de esta alternativa es que no se necesitan de dispositivos con Sistema de Posicionamiento Global (GPS) para que funcionen, sino que hoy en día los teléfonos inteligentes y/o dispositivos que cuentan con un GPS ayudaría a mandar la ubicación en tiempo real del autobús.

### 2.2.3 Desventajas

Dado que los teléfonos celulares de los usuarios serán los que manden información a los servidores se requiere que al menos uno de ellos tenga la aplicación descargada, la tenga abierta en el autobús y que tenga acceso a internet, también aquellos usuarios que no cuenten con un teléfono inteligente no podrán acceder a dicho servicio. O bien que los conductores tengan internet en sus teléfonos para que puedan mandar la información a los servidores y pueda ser analizada.

## 2.4 Parámetros para el cálculo del tiempo estimado.

Para el cálculo de tiempos estimados se llevaron a cabo mediante google maps<sup>1</sup>, el cual nos da un tiempo estimado basado en tres factores, tráfico que se define como la cantidad de vehículos que pasan por un determinado punto en un cierto intervalo de tiempo, velocidad se define como una razón de movimiento en distancia por

---

<sup>1</sup> Es una herramienta de búsqueda de ubicaciones que permite geolocalizar un punto concreto, calcular rutas, encontrar los lugares de interés más cercanos o ver la apariencia de un lugar a pie de calle a través de [Google Street View](#). Fue desarrollada en 2005, inicialmente para Internet Explorer y Mozilla Firefox en PC. Sin embargo, su uso se ha generalizado de la mano de los smartphones y ha sido entonces cuando ha pasado a convertirse en un elemento fundamental para hacer SEO local.

unidad de tiempo y distancia se define como la longitud total de una cierta trayectoria de un objeto entre dos puntos. Sabemos que el autobús tiene un carril exclusivo dentro de CU y consideremos como condiciones ideales cuando no hay tráfico donde la velocidad promedio del autobús es de 40 km/h. Con esto se puede hacer un cálculo estimado del tiempo. La información faltante está relacionada con el tiempo que se hace en cada una de las estaciones donde se realiza una parada.

Una alternativa para hacer un cálculo estimado del tiempo del autobús es mediante el sistema que se planteó anteriormente con los GPS tomando en cuenta los mismos factores que con google maps.

Para reducir la incertidumbre de los tiempos también se tomaron de forma experimental, los datos con los que se trabajo en este proyecto son inventados, ya que por el momento no estaba disponible el servicio de Pumabús.

En este capitulo se hizo uso de las siguientes tecnologías:

- Teléfonos inteligentes
- Internet
- GPS
- Servidores

Los cuales son indispensables para poder cumplir con uno de los objetivos del proyecto que es hacer el calculo del tiempo que tardará en llegar el autobús a alguna parada en tiempo real.

Se dieron a conocer las diferentes sistemas para el calculo del tiempo que tardará en llegar el autobús a alguna parada en tiempo real, además se vieron ventajas y desventajas de los sistemas propuestos. Por último se describió el modelo que se uso para la recolección de datos que es el cliente/servidor.

## Capítulo 3

### Desarrollo de aplicación

En este capítulo se definirá el diseño y desarrollo de la aplicación, también se verá agrosomodo el lenguaje con el que fue programada y se verán todos los recursos utilizados.

#### 3.1 Teconología Xamarin Forms

##### 3.1.1 ¿Qué es Xamarin Forms?

Es una plataforma que permite el desarrollo de aplicaciones móviles para los dos principales sistemas operativos que son *Android* e *iOS* aunque también permite el desarrollo para *Windows*. Un recuso muy atractivo que ofrece *Xamarin* es que el desarrollo se puede hacer desde un único código base compartido.

De igual manera permite la creación de interfaces de usuario en el lenguaje *XAML* con código subyacente en *C#*. Para realizar la aplicación se requiere la descarga de *Visual Studio*.

##### 3.1.2 ¿Qué es Visual Studio?

*“Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .Net, F#, Java, Python, Ruby y PHP.*

*Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros”*

**Fuente:** Microsoft Visual Studio (25 de agosto de 2022). En Wikipedia. [https://es.wikipedia.org/w/index.php?title=Microsoft\\_Visual\\_Studio&oldid=1455872](https://es.wikipedia.org/w/index.php?title=Microsoft_Visual_Studio&oldid=1455872)

### 3.2 Desarrollo de la Aplicación

Con base a todo lo mencionado en los dos capítulos anteriores, se procede al desarrollo de la aplicación. La aplicación se compone de 3 elementos principales, un menú de búsqueda, inicio de sesión y un menú de hamburguesa en el cual se podrán apreciar cada una de las rutas y estaciones del pumabus. Esto se explicará más adelante.

Es importante mencionar que en la actualidad los principales sistemas operativos de teléfonos inteligentes son *Android* e *iOS*, razón por la cual se eligió esta plataforma para el desarrollo de la aplicación, cabe resaltar que permite el desarrollo para ambas plataformas, además con la peculiaridad de programar con código compartido, de esta manera la aplicación adquiere un aspecto nativo. Con nativo se hace referencia a que es como si la aplicación hubiese sido desarrollada alguna plataforma específicamente para desarrollo de Android como React, Angular, Kotlin, etcétera. por mencionar algunos, de igual manera sucede para iOS pero en este caso se tiene Xcode.

Se empieza con un proyecto en blanco en *Visual Studio*.

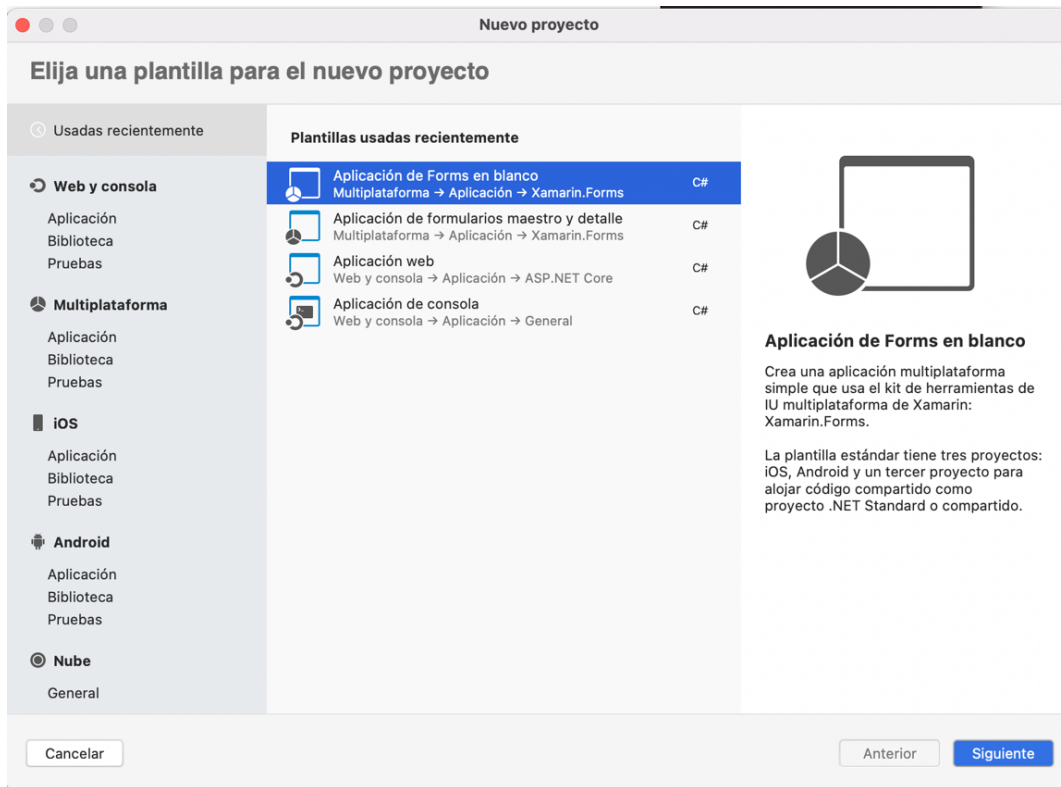
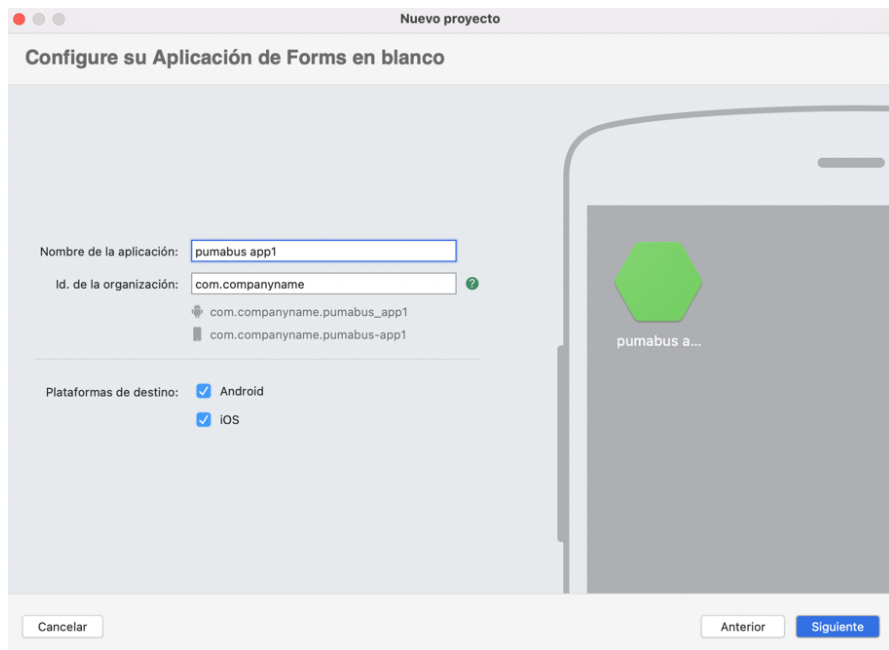


Figura 3.2.1 Creación de proyecto

En la figura 3.2.1, del lado izquierdo se muestran diferentes apartados, se selecciona en el apartado de multiplataforma la opción de “Aplicación” seguido a esto, se observa un menú con 4 opciones, se selecciona “Aplicación de Forms en blanco”, en seguida se hace selección en siguiente.

Luego se muestra una pantalla donde tendremos que nombrar el proyecto.





*Figura 3.2.2 Selección de nombre de proyecto.*

La figura 3.2.2 muestra dos campos, el primero es donde pondremos el nombre del proyecto, la parte inferior muestra 2 opciones que dicen Android e iOS, se seleccionaron ambos debido a que la aplicación es multiplataforma. Por otra parte del lado derecho se muestra una simulación de como se vería el nombre en un teléfono inteligente.

Una vez creado el proyecto se muestra una pantalla con la estructura de archivos que conforman al proyecto.

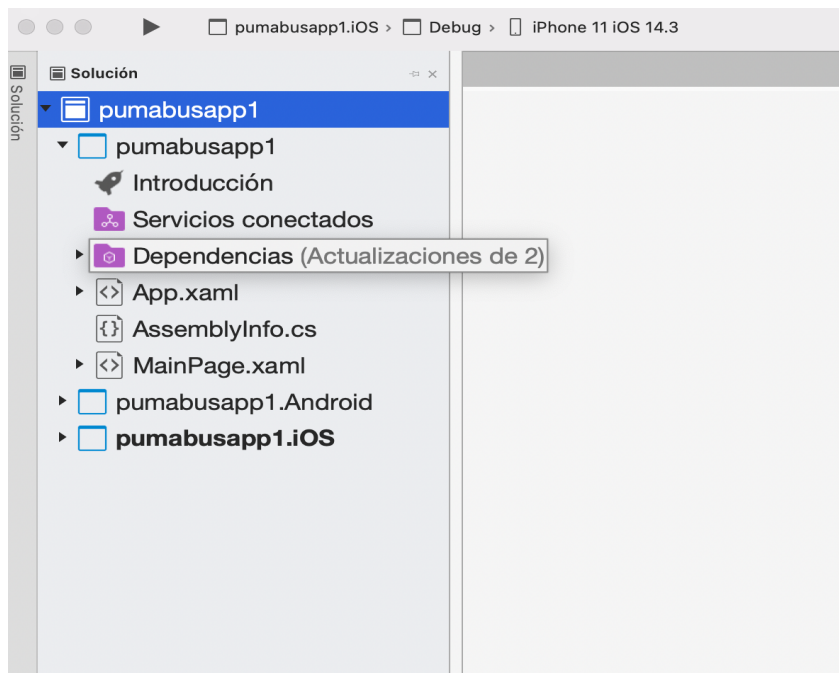


Figura 3.2.3 Pantalla de inicio de aplicación.

En la figura 3.2.3 del lado izquierdo en la parte de la solución se muestra una pantalla con la estructura de archivos que conforman el proyecto de inicio, el primer campo hace referencia al nombre del proyecto, que en este caso es “pumabusapp1”, seguido a esto se muestra una carpeta con el mismo nombre del proyecto que contiene los archivos que nos da por defecto la solución. Por último se muestran dos carpetas que hacen referencia a los sistemas operativos Android e iOS.

En el recuadro superior se muestran 4 elementos que son, primero un triángulo que servirá para correr la aplicación, segundo el nombre de la aplicación con la terminación .iOS o .Android, que indica en cual sistema operativo haremos las pruebas, tercero una opción donde se puede configurar entre probar en un simulador, descargar a un dispositivo o bien lanzar directo a la App Store en caso de iOS o a play Store en caso de Android. Por último se podrá elegir entre diferentes dispositivos para llevar a cabo las pruebas necesarias.

Para empezar con el desarrollo de la aplicación, primero se va a agregar un archivo *.xaml* al cual vamos a llamar *masterPage* y una vez agregado este archivo seleccionamos *MainPage* y se escribe el código correspondiente para obtener el botón que dará el acceso al menú lateral en el cual podemos encontrar todas las rutas del Pumabús así como un botón de más información que nos redirigirá a la página oficial del Pumabús. Después agregamos otro archivo *.xaml* al cual denominamos *Tercera*, el cual será la página principal de la aplicación.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <MasterDetailPage xmlns="http://xamarin.com/schemas/2014/
3     xmlns:x="http://schemas.microsoft.com/A
4     xmlns:local="clr-namespace:MasterDetail
5     x:Class="MasterDetailPageNavigation.Ma:
6     <MasterDetailPage.Master>
7         <local:MasterPage x:Name="masterPage" />
8     </MasterDetailPage.Master>
9     <MasterDetailPage.Detail>
10        <NavigationPage>
11            <x:Arguments>
12                <local:Tercera />
13            </x:Arguments>
14        </NavigationPage>
15    </MasterDetailPage.Detail>
16 </MasterDetailPage>
17
```

Código 3.2.1 *MainPage.xaml*

En la figura 3.2.1, se muestra el código de *MainPage.xaml* en la línea 1 se hace referencia a que usaremos cualquier tipo de notación incluyendo caracteres especiales, en la línea 2 declaramos una página con nombre “*MasterDetailPage*” con el fin de obtener el menú lateral o también conocido como menú de hamburguesa, de la línea 2 hasta la 5, configuramos permisos de Microsoft. De la línea 6 hasta la 16 se configura “*MasterDetailPage*”, en la línea 7 configuramos *masterPage* que es la página que contendrá el menú de hamburguesa, en la línea 12 declaramos la página “*Tercera*”, la cual será la vista principal de la aplicación.

En el código masterPage se declaran cada una de las rutas y se nombran con el fin de que se muestren en el menú de hamburguesa al igual que el botón de más información.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4      xmlns:local="using:MasterDetailPageNavigation"
5      x:Class="MasterDetailPageNavigation.MasterPage"
6      Padding="0,0,0,40"
7      IconImageSource="hamburger.png"
8      Title="Rutas Pumabus"
9  >
10
11  <StackLayout>
12      <ListView x:Name="listView" x:FieldModifier="public" BackgroundColor="CadetBlue" SeparatorColor="Yellow" >
13
14          <ListView.ItemsSource>
15              <x:Array Type="{x:Type local:MasterPageItem}">
16
17                  <local:MasterPageItem Title="Metro Universidad" IconSource="metrou.png" />
18                  <local:MasterPageItem Title="Ruta1" IconSource="ruta1.png" TargetType="{x:Type local:ContactsPage}" />
19                  <local:MasterPageItem Title="Ruta2" IconSource="ruta2.png" TargetType="{x:Type local:TodoListPage}" />
20                  <local:MasterPageItem Title="Ruta3" IconSource="ruta3.png" TargetType="{x:Type local:ReminderPage}" />
21                  <local:MasterPageItem Title="Ruta4" IconSource="ruta4.png" TargetType="{x:Type local:Ruta_4}" />
22                  <local:MasterPageItem Title="Ruta5" IconSource="ruta5.png" TargetType="{x:Type local:Ruta5}" />
23
24                  <local:MasterPageItem Title="Estadio" IconSource="ruta1.png" />
25                  <local:MasterPageItem Title="Ruta6" IconSource="ruta6.png" TargetType="{x:Type local:Ruta6a}" />
26                  <local:MasterPageItem Title="Ruta7" IconSource="ruta7.png" TargetType="{x:Type local:Ruta7}" />
27                  <local:MasterPageItem Title="Ruta8" IconSource="ruta8.png" TargetType="{x:Type local:Ruta8}" />
28
29                  <local:MasterPageItem Title="Metrobús CU" IconSource="metrobus.png" />
30                  <local:MasterPageItem Title="Ruta9" IconSource="ruta9.png" TargetType="{x:Type local:Ruta9}" />
31                  <local:MasterPageItem Title="Ruta10" IconSource="ruta10.png" TargetType="{x:Type local:Ruta_10}" />
32                  <local:MasterPageItem Title="Ruta11" IconSource="ruta11.png" TargetType="{x:Type local:Ruta11}" />
33
34                  <local:MasterPageItem Title="Ruta Expres" IconSource="regresa.png" />
35                  <local:MasterPageItem Title="Ruta13" IconSource="ruta13.png" TargetType="{x:Type local:Ruta13}" />
36                  <local:MasterPageItem Title="Más información" IconSource="info.png" TargetType="{x:Type local:Ruta12}" />
37
38
39
40
41          </x:Array>
42      </ListView.ItemsSource>
43      <ListView.ItemTemplate>
44          <DataTemplate>
45              <ViewCell>
46                  <Grid Padding="5,10">
47                      <Grid.ColumnDefinitions>
48                          <ColumnDefinition Width="30"/>
49                          <ColumnDefinition Width="*" />
50                      </Grid.ColumnDefinitions>
51                      <Image Source="{Binding IconSource}" />
52                      <Label Grid.Column="1" Text="{Binding Title}" />
53                  </Grid>
54              </ViewCell>
55          </DataTemplate>
56      </ListView.ItemTemplate>
57  </ListView>
58 </StackLayout>
59 </ContentPage>
60
61
62
63

```

Código 3.2. 3 Archivo MasterPage.xaml

En el código 3.2.3 se muestra el código de Archivo MasterPage.xaml, en la línea 6 se menciona "Padding" lo que sirve para tener una disposición de los elementos que agreguemos, como se muestra se compone de 4 elementos, que van de cero a cien

y separados por comas. En la línea 7 “IconImageSource” hace referencia a la image que se utilizara para representar el menú de hamburguesa. En la línea 8 “Title” ayuda a definir el nombre que le pondremos a la página.

De la línea 11 a la línea 41 se muestra la configuración de cada una de las rutas y del botón de más información. Se nombró cada una de las rutas y se agregaron los archivos .xaml correspondientes. Están separadas en 4 categorías, las que pasan por “metro universidad” las que pasan por “estadio”, las que pasan por “metrobus CU” y la ruta exprés. De la línea 42 a la 62 se muestran las configuraciones en cuanto a la disposición de los elementos.

### 3.2.1 Rutas y botón de más información.

En esta sección se muestra el desarrollo de una de las rutas del Pumabús, el proceso para las demás es análogo, se analiza la ruta 4 la cual empieza en “Química Conjunto D y E” y termina en “Facultad de Ciencias”.

Como se observa están nombradas cada una de las paradas y en la parte inferior muestra un mensaje de tiempo estimado de llegada del Pumabús que dice “llega en...” con el fin de que el usuario al ver la estación de su interés tuviera conocimiento de un tiempo estimado de llegada del Pumabús. Por otra parte se implementan dos TextCell (celdas de texto) la primera con la opción de “ver mapa” que al ser seleccionada muestra un mapa estático de la ruta y la segunda opción es abrir Google Maps, con el fin de que el usuario pueda tener una mejor experiencia al usar la aplicación.

El código 3.2.4 muestra el procedimiento para la construcción de la ruta 4

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4             xmlns:d="http://xamarin.com/schemas/2014/forms/design"
5             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6             xmlns:local="clr-namespace:MasterDetailPageNavigation"
7             mc:Ignorable="d"
8             x:Class="MasterDetailPageNavigation.Ruta_4"
9             Title="Ruta 4"
10            Padding="20,40,20,0"
11            BackgroundColor="SaddleBrown">
12
13     <ContentPage.ToolbarItems>
14
15         <ToolbarItem Text="Menú"
16                     Command="{Binding NavigateCommand}"
17                     CommandParameter="{x:Type local:Tercera}" />
18
19
20
21     </ContentPage.ToolbarItems>
22
23
24     <ScrollView>
25     <TableView Intent="Menu" IsEnabled="True" BackgroundColor="WhiteSmoke">
26         <TableRoot>
27             <TableSection>
28                 <TextCell Text="Metro Universidad"
29                         Detail="llega en... "
30                         />
31
32                 <TextCell Text="Química Conjunto D y E"
33                         Detail="llega en... "
34                         />
35
36                 <TextCell Text="Facultad de Ciencias Alumnos"
37                         Detail="llega en... "
38                         />
39
40                 <TextCell Text="Facultad de Contaduría y Administracion"
41                         Detail="llega en... "
42                         />
43
44                 <TextCell Text="Escuela de Trabajo Social"
45                         Detail="llega en... "
46                         />
47
48                 <TextCell Text="Estadio de Practicas"
49                         Detail="llega en... "
50                         />
51
52                 <TextCell Text="Campos de Futbol I"
53                         Detail="llega en... " />
54
55                 <TextCell Text="Jardín Botánico"
56                         Detail="llega en... "
57                         />
58
59                 <TextCell Text="Campos de Futbol II"
60                         Detail="llega en... "
61                         />
62                 <TextCell Text="Metrobús CU"
63                         Detail="llega en... "
64                         />
65                 <TextCell Text="Educación a Distancia"
66                         Detail="llega en... "
67                         />
68                 <TextCell Text="D.G.T.I.C"
69                         Detail="llega en... "
70                         />
71                 <TextCell Text="Facultad de Ciencias"
72                         Detail="llega en... "
73                         />
74

```

```

75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102

```

```

<TextCell Text="ver mapa"
  TextColor="#8B4513" SaddleBrown"
  Command="{Binding NavigateCommand}"
  CommandParameter="{x:Type local:Ruta_4a}"
/>
<TextCell Text="ver mapa fijo"
  TextColor="#8B4513" SaddleBrown"
  Command="{Binding NavigateCommand}"
  CommandParameter="{x:Type local:Ruta_4b}"
/>

<TextCell Text="Información"
/>

</TableSection>
</TableRoot>
</TableView>

</ScrollView>

</ContentPage>

```

Código 3.2.4 Código de Ruta4.xaml

En el código 3.2.4 se muestra como se configuró cada una de las paradas de la Ruta 4, en la línea 11 “BackgroundColor” se definió el color de fondo, en este caso se decidió por un café oscuro dado que la ruta 4 lleva ese color. De la línea 13 a la línea 21, se configuró un botón llamado “menú” que al hacer clic se regresa a la página principal de la aplicación.

De la línea 24 a la 74, en formato de tabla nombramos cada una de las paradas. De la línea 76 a la 101, se definen los mapas a mostrar y el botón de más información.

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <ContentPage
3      xmlns="http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="MasterDetailPageNavigation.Ruta_4b">
6      <ContentPage.Content>
7          <Image Source="ruta4m.png" >
8
9          </Image>
10     </ContentPage.Content>
11 </ContentPage>
12

```

Código 3.2.5 Código de mapa estático

En el código 3.2.5 se muestra la realización del mapa estático. En la línea 7 “Image Source” ayuda a la implementación del mapa que es una imagen en formato png. La implementación de este mapa.

El mapa de Google Maps requiere la instalación de un paquete “Nugget” para ambos sistemas operativos, en el caso de Android se necesita adquirir un permiso especial desde google para poder mostrar el mapa, para ello solo se requiere declarar el tipo de mapa que se desea.

Se procede a analizar el código del mapa de Google Maps.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4             xmlns:maps="clr-namespace:Xamarin.Forms.Maps;assembly=Xamarin.Forms.Maps"
5             x:Class="MasterDetailPageNavigation.Ruta_4a"
6             Title="Mapa Ruta4">
7
8      <StackLayout Margin="10">
9          <maps:Map x:Name="map" />
10         <Slider Margin="20,0,20,0"
11              Maximum="18"
12              Minimum="1"
13              Value="12"
14              ValueChanged="Slider_ValueChanged" />
15         <StackLayout Orientation="Horizontal"
16              HorizontalOptions="Center"
17              Spacing="30">
18             <Button Text="Street"
19                  Clicked="Button_Clicked" />
20             <Button Text="Satellite"
21                  Clicked="Button_Clicked" />
22             <Button Text="Hybrid"
23                  Clicked="Button_Clicked" />
24         </StackLayout>
25     </StackLayout>
26 </ContentPage>
```

Código 3.2.6 Código para Google Maps

El código 3.2.6 de la línea 8 a la 14, se asignan propiedades para que el mapa cuadre dentro de la pantalla del dispositivo, de la línea 15 a la 27, se declaran 3 botones para tener 3 tipos de vistas diferentes, uno de “calle”, otro de “satelite” y el último “hibrido” que es una combinación de los dos anteriores. Un detalle a resaltar es que cuando se muestra el mapa abre directo sobre Ciudad Universitaria.

Se muestra el *codebehind* (código detrás del código) del Código para Google Maps.



```

1  using Xamarin.Forms;
2  using System;
3  using Xamarin.Forms.Maps;
4
5  namespace MasterDetailPageNavigation
6  {
7      public partial class Ruta_4a : ContentPage
8      {
9          public Ruta_4a()
10         {
11             InitializeComponent();
12             map.MoveToRegion(
13                 MapSpan.FromCenterAndRadius(
14                     new Position(19.328888888, -99.18722222), Distance.FromKilometers(1)));
15         }
16     }
17
18     void Slider_ValueChanged(System.Object sender, Xamarin.Forms.ValueChangedEventArgs e)
19     {
20         double zoomLevel = e.NewValue;
21         double latlongDegrees = 360 / (Math.Pow(2, zoomLevel));
22         if (map.VisibleRegion != null)
23         {
24             map.MoveToRegion(new MapSpan(map.VisibleRegion.Center, latlongDegrees, latlongDegrees));
25         }
26     }
27
28     void Button_Clicked(System.Object sender, System.EventArgs e)
29     {
30         Button button = sender as Button;
31         switch (button.Text)
32         {
33             case "Street":
34                 map.MapType = MapType.Street;
35                 break;
36             case "Satellite":
37                 map.MapType = MapType.Satellite;
38                 break;
39             case "Hybrid":
40                 map.MapType = MapType.Hybrid;
41                 break;
42         }
43     }
44 }

```

*Código 3.2.7 Opciones de vista de mapa.cs*

En el código 3.2.7 de la línea 1 a la 3 se muestran las propiedades utilizadas, en específico la línea 3 hay una propiedad para mostrar el mapa. De la línea 5 a la 14 se muestra el procedimiento para poner las coordenadas, en la línea 14 se definieron las coordenadas de Ciudad Universitaria. De la línea 18 a la 24 se declaró un “zoom” o un acercamiento el cual tiene la función de que el usuario pueda alejar o acercar el mapa. Por último de la línea 28 a la 41 se tiene la declaración de los botones de “calle”, “satelite” e “hibrido”.

Para el botón de más información, se agregó un nuevo archivo .xaml con nombre “ruta 12” pero para alcanzar la función que se desea se accede al codebehind que tiene la terminación .xaml.cs.

```
1 using System;
2 using System.Windows.Input;
3 using Xamarin.Forms;
4
5 namespace MasterDetailPageNavigation
6 {
7     public partial class Ruta12 : ContentPage
8     {
9         public Ruta12()
10        {
11            InitializeComponent();
12            var browser = new WebView();
13            browser.Source = "https://www.dgsgm.unam.mx/pumabus";
14            Content = browser;
15
16            NavigateCommand = new Command<Type>(async (Type pageType) =>
17            {
18                Page page = (Page)Activator.CreateInstance(pageType);
19                await Navigation.PushAsync(page);
20            });
21
22            BindingContext = this;
23        }
24
25        public ICommand NavigateCommand { private set; get; }
26    }
27 }
```

*Código 3.2.8 Código de WebView*

En el código 3.2.8 de la línea 5 a la 26, se muestra como se hizo la referencia para enlazar el botón con la página web, en la línea 13 con el comando “browser.Source” se pega la liga de la página oficial del Pumabús.

El inicio de sesión tiene como función principal que las personas que pertenecen a la comunidad de la Universidad se registren con la finalidad de tener un mejor control de usuarios.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4             x:Class="MasterDetailPageNavigation.Sesion"
5             xmlns:d="http://xamarin.com/schemas/2014/forms/design"
6             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
7             xmlns:local="clr-namespace:MasterDetailPageNavigation"
8             mc:Ignorable="d">
9     <ContentPage.Content>
10        <StackLayout>
11            <Label Text="Username" />
12            <Entry x:Name="usernameEntry" Placeholder="username" />
13            <Label Text="Password" />
14            <Entry x:Name="passwordEntry" />
15            <Button Text="Login" />
16            <Label x:Name="messageLabel" />
17            <Button Text="Continuar como invitado"
18                    VerticalOptions="End"
19                    HorizontalOptions="Center"/>
20        </StackLayout>
21    </ContentPage.Content>
22 </ContentPage>
23
```

*Código 3.2.9 Código de inicio de sesión*

En el código 3.2.9, de la línea 9 a la 22 se define el inicio de sesión, en la línea 12 se define el usuario mientras en la línea 13 se define la contraseña del mismo, seguido en la línea 15 se define el botón de “login” para acceder, por otro lado en la línea 17 se define el ingreso como invitado.

La barra de búsqueda, tiene como función buscar cualquier estación del Pumabús para poder trasladarse de una estación a otra, la aplicación mostrará el número del camión a tomar así como la ruta más corta para llegar al destino. Se agrega en el archivo Tercera un “searchbar” o barra de búsqueda, se debe agregar una base de datos con cada una de las estaciones y rutas del Pumabús, además el algoritmo para calcular la ruta más corta.

Se muestra el código del “searchbar”

```
<StackLayout>
  <SearchBar
    Placeholder="¿A donde quieres ir?"
    TextColor=■ "Black"
    BackgroundColor=■ "LightBlue"
    CancelButtonColor=■ "Black"/>
```

*Código 3.2.10 Función barra de búsqueda*

En el código 3.2.10 el "Placeholder" es lo que se muestra en la barra de búsqueda, las siguientes propiedades son de color de texto, color de fondo y el color del botón de "Cancel".

### 3.3 Algoritmo para la ruta más corta

Para tener una barra de búsqueda que opere de manera eficaz se necesita recurrir a un algoritmo para encontrar la ruta más corta, con el fin de que el usuario cuando desee trasladarse de un punto cualquiera a un punto B utilice la menor cantidad de autobuses y vaya por la ruta más corta, este algoritmo se explicará más adelante a detalle.

#### 3.3.1 Aplicación de algoritmo

En esta sección se expone la implementación del algoritmo para calcular la ruta más corta entre dos puntos dentro de CU para lo cual se tomará como referencia la distancia que hay entre cada parada del autobús y así poder calcular la ruta más corta entre cada parada. Se utilizará lo antes mencionado en el marco teórico del capítulo 1.

Se expone un ejemplo de cómo funciona el algoritmo, calculando la ruta más corta entre dos paradas del autobús tomadas aleatoriamente y el funcionamiento es análogo para todas las demás.

Las paradas de autobús en las cuales se basa este ejemplo serán la Facultad de Ciencias como origen y la Facultad de Economía como destino. Se muestra en el siguiente diagrama las distancias que hay entre cada parada para así poder aplicar el algoritmo de la ruta más corta.

El funcionamiento del algoritmo es de la siguiente forma:

Paso 1: validar si por la ruta en la que se viaja se llega al destino final, si es así se suma la distancia para verificar que sea la ruta más corta, de lo contrario se avanza a la siguiente parada de autobús, posteriormente se verifica la conexión entre cada parada de autobús y se verifica si alguna ruta llega al destino final de lo contrario se repite el proceso.

Paso 2: Ubicar las posibles rutas del origen al destino y sumar las distancias para verificar cual es la ruta más corta.

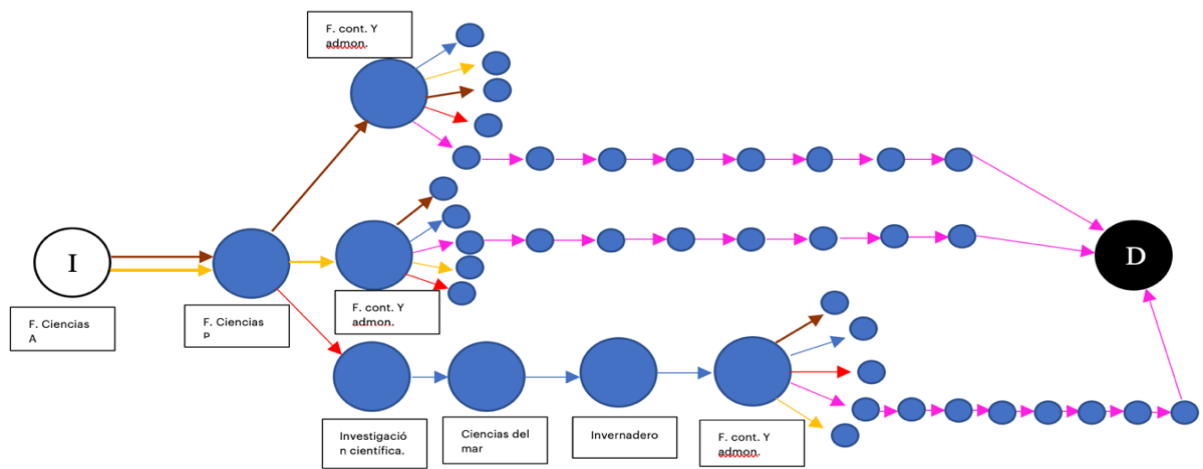
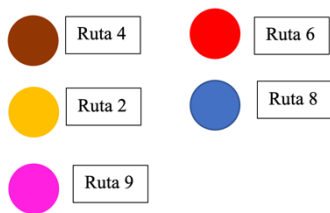


Figura 3.3.12 Diagrama de la ruta más corta



En el ejemplo anterior ver (Figura 3.20) se tomó como inicio la Facultad de Ciencias alumnos y como destino final la Facultad de Economía, lo primero que se verificaron las rutas pasan por esa parada y en este caso son (la ruta 2 y la ruta 4). El algoritmo inicia, es indistinto la ruta que se tome debido que las 2 van a la misma parada, por lo tanto se avanza a la siguiente parada.

En la siguiente parada se verifican las rutas que pasan por esa parada, son la ruta 4, la ruta 2 y se agrega la ruta 6, sin embargo con esa ruta no se puede llegar al destino, entonces el algoritmo verifica la siguiente parada de autobús.

Se repite el proceso hasta encontrar alguna parada de autobús que nos lleve al punto destino, lo que sigue es calcular la ruta más corta.

En este ejemplo se obsecua que es indistinto tomar la ruta 2 ó la 4 ya que ambas tienen la misma distancia, aquí el factor que hará la diferencia es el autobús que llegue primero al origen.

En este capítulo se vio detalladamente el diseño y desarrollo de la aplicación, se mostraron los códigos de todos los elementos que componen la aplicación. Se mostró la aplicación del algoritomo, se vio un ejemplo concreto para su mejor entendimeinto.

Las tecnologías utilizadas son:

- Xamarin Forms
- Visual Studio
- Google Mapas

Se eligió realizar el proyecto con Xamarin Forms en Visual Studio ya que las características que ofrece son lo más viable para la realización del proyecto y para cumplir con los objetivos antes mencionados. De igual forma con Google Maps, se obtó por su uso gracias a que es una herramienta que permite la búsqueda de ubicaciones, además permite geolocalizar un punto en concreto, que fue lo que permitió tener el mapa interactivo localizado en Ciudad Universitaria.

## Capítulo 4

### Funcionamiento de la aplicación y resultados

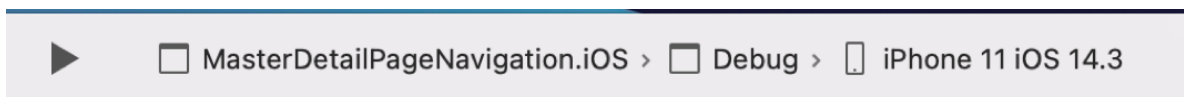
En este capítulo se verá el funcionamiento de la aplicación, se harán simulaciones en ambos sistemas operativos, para conocer el funcionamiento de ésta, de igual forma se verán los resultados de todo el proyecto.

#### 4.1 Funcionamiento de la aplicación y resultados.

La aplicación se implementará en un simulador para ambos sistemas operativos IOS e Android, se verá la aplicación terminada y todas sus funciones.

##### 4.1.1 Implementación en iOS.

La simulación se llevará a cabo en un iPhone 11 con sistema operativo iOS 14.3 como se muestra en la siguiente figura (Figura 4.1).



*Figura 4. 1 Simulación en Iphone 11*

Para comenzar la simulación se debe hacer clic en el botón de *Play*, está marcado con color azul. Una vez realizado esto se descargará la App en el dispositivo e iniciará.





Figura 4.2 Pantalla SplashScreen.



Figura 4.3 Pantalla principal.

En la Figura 4.2 se aprecia la “splash screen” o pantalla de inicio que se muestra al iniciar la aplicación, cuenta con 3 elementos, un título y dos imágenes.

La Figura 4.3 muestra la pantalla principal una vez unificada la aplicación, en la parte superior dentro el recuadro rojo se señala en la parte izquierda el “botón de hamburguesa” que es el cual permite el acceso a todas las rutas y botón de más información, en la parte central se indica el nombre de la aplicación “PUMABUS APP”. Por último, del lado derecho se observa el botón de “inicio de sesión”.

En la parte central de la pantalla marcada con un rectángulo azul se observan dos elementos. El primero es la “barra de búsqueda” y el segundo es la parte texto y debajo una imagen la cual es el “logotipo del Pumabús”.

Por último, en la parte inferior de la pantalla marcada con un rectángulo naranja se observa una imagen de todas las rutas para que así el usuario pueda ver todas las rutas juntas.

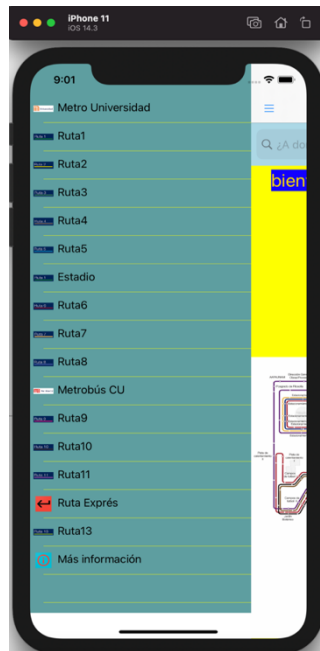


Figura 4.4 Menú Lateral

En la Figura 4.4 se tienen las doce rutas, además se tienen ciertas celdas en las que se indica las rutas que pasan por Metro Universidad, por el Estadio de CU y por el Metrobús, se puede elegir cada una de ellas y mostrará cada una de las paradas de autobús además de incluir un botón para mostrar un mapa interactivo y uno fijo, también se cuenta con botón que de forma específica mostrará información.

Cuando se seleccione alguna de las rutas, se podrá observar cada una de las paradas de autobús, se muestra la ruta 4 como ejemplo para comprobar todo lo mencionado con anterioridad.

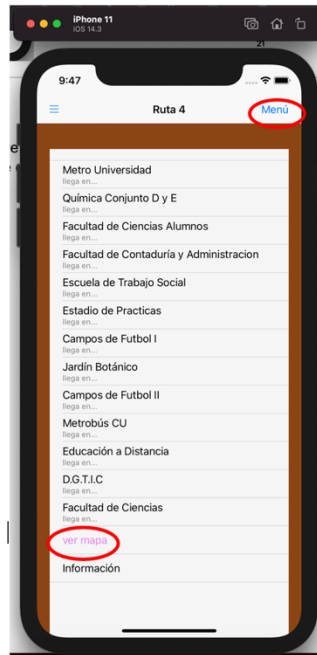


Figura 4.5 Ruta 4

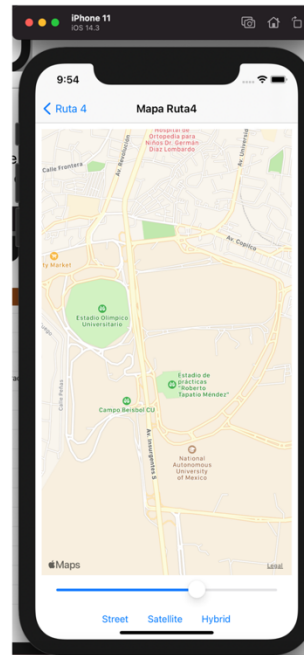


Figura 4.6 Mapa Ruta 4

En la Figura 4.5 en la parte superior derecha se muestra un botón que dice “menú” el cual tiene la función de volver a la página principal. En la Figura 4.6 se muestra uno de los dos mapas que se pueden abrir, en este caso se muestra el mapa interactivo que está ubicado en el campus CU. En la parte inferior se encuentran los tres botones descritos en la parte del desarrollo.

La Figura 4.7 muestra el botón de más información, como se explicó en el capítulo anterior, va a redirigir a la página del PUMABUS.

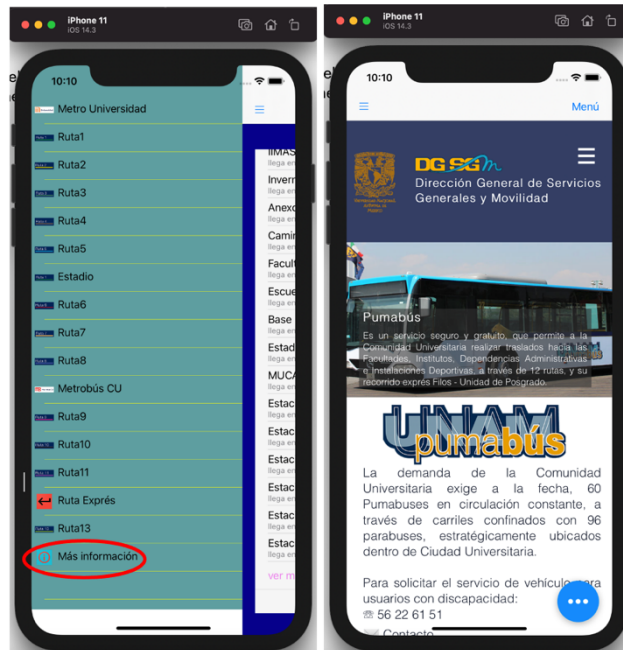


Figura 4.7 Botón más información. Figura 4.8 Página más información

En la Figura 4.8 se muestra la página oficial del pumabús, en caso que el usuario desee consultar alguna información adicional.

Por último se muestra la página principal, en la parte superior izquierda vemos el “inicio de sesión”. Para que esto tenga un funcionamiento óptimo se requiere servidores para manejar bases de datos.



Figura 4.9 Botón de Inicio de Sesión      Figura 4.10 Inicio de Sesión

En la Figura 4.10 se observa el inicio de sesión, en la parte superior se observan 4 campos, el primero “Username” (nombre de usuario) y enseguida un celda para escribirle, el tercero “Password” (contraseña) y enseguida un celda para escribirle. Seguido se muestra el botón de “Login” (acceso) y por último un botón de “Continuar como invitado”.

Se omite la simulación en Android dado que es análoga a la de iOS.

## Conclusiones

Actualmente el uso de las aplicaciones son de gran utilidad, hoy en día su uso se ha vuelto casi indispensable, ya sea con fines de comunicación, entretenimiento, transporte o servicio de comida por mencionar algunos, en el caso del proyecto recae en el área de transporte.

Durante el desarrollo de este trabajo se cumplió con los objetivos planteados de manera satisfactoria:

- Se logró realizar una aplicación que ayudará al sistema de transporte interno Pumabús apoyado por el sistema de control de usuarios para así veitar conglomeraciones, desafortunadamente no se pudo probar dentro de las instalaciones de Ciudad Universitaria dadas las condiciones en las que actualmente nos encontramos, y es incierto cuando se podrá probar ya que no hay fecha exacta de cuando se regrese a clases o actividades presenciales.
- La aplicación brinda a los usuarios tiempos estimados de la llegada del pumabus, además de ayudarles a moverse mejor dentro de Ciudad Universitaria en el Pumabus, esto al la aplicación tener una barra de búsqueda con la que podrán ingresar su punto de origen y su destino, seguido la aplicación indicará cual es la ruta más corta para llegar a su destino también podrán visualizar las 12 rutas con cada una de las paradas.
- Utilizando el algoritmo de la ruta más corta, la aplicación proporciona a los usuarios una herramienta de búsqueda la cual les permite trasladarse de una Facultad a otra brindándoles la mejor ruta para llegar a su destino.
- Permitirá predecir los tiempos estimados del autobús en cada parada, ayudando a que los usuarios tengan conocimiento del mismo y tomen todas las precauciones necesarias dadas las condiciones actuales en las que nos encontramos hoy en día.
- Ayudará a los usuarios a tener conocimiento del tiempo estimado de la llegada del autobús en cada parada, esto es importante para que así los usuarios tomen la decisión si esperan el autobús o recurren a algún otro medio de

transporte, además de que de esta forma se reduce el número de personas por parada.

Antes que vivieramos este periodo de pandemia Covid-19 la población en general de la Universidad podían hacer uso de cualquier parada del servicio Pumabús sin considerar medidas preventivas como son la sana distancia, cubrebocas, gel antibacterial etc... entre otra medidas, sin embargo en las condiciones actuales esto no es posible. Con la aplicación es posible ofrecer un escenario de prevención donde los usuarios de toda la Universidad puedan evitar conglomeraciones por desconomiento de la hora exacta de llegada del Pumabús. En el futuro, la aplicación permitirá prevenir especialmente conglomeraciones en las Facultades con mas estudiantes así como lograr un eficiente uso del tiempo para dicha población.

Haber estudiado una licenciatura en matemáticas me sirvió de forma consoderable para la realización del proyecto ya que al tener amplios conocimientos principalmente en lógica matemática, algebra y cálculo entre otros, me resultó más sencillo adentrarme en el mundo de la programación ya que considero que me facilitó la comprensión y escritura de código.

## Bibliografía

- [1] O. Q. Bibiana, «Repositorio de la Facultad de Ingeniería,» 30 08 2005. [En línea]. Available: <http://www.ptolomeo.unam.mx:8080/xmlui/handle/132.248.52.100/539>. [Último acceso: 2022].
- [2] J. J. J. G. Mokhtar S. Bazaraa, Programación lineal y flujo en redes La imagen de portada del libro no está disponible Autores: Mokhtar S. Bazaraa, John J. Jarvis Gutiérrez Editores: México D.F. [etc. : Limusa, 1981 Año de publicación: 1981 País: México Idioma: español ISBN: 968-18-132, México: Limusa, 1981.
- [3] I. F. d. I. Mota, Apuntes de Teoría de Redes, Mexico: Facultad de Ingeniería, UNAM, 1999.
- [4] D. G. d. Planeación, «Agenda Estadística 2020 UNAM,» 2020. [En línea]. Available: <https://www.planeacion.unam.mx/Agenda/2020/pdf/Agenda2020.pdf>.
- [5] S. D. G. D. P. CÉSAR MIGUEL BARRIENTOS ROBLES, «TESIS UNAM,» 2015. [En línea]. Available: <http://132.248.9.195/ptd2015/octubre/0736890/Index.html>. [Último acceso: abril 2021].
- [6] Soto, M. (2013) Modelación de hiper rutas mínimas multimodales del transporte [4] [back4app.com/es/blog/back4app-com-es-las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/](https://back4app.com/es/blog/back4app-com-es-las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/)
- [7] [back4app.com/es/blog/back4app-com-es-las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/](https://back4app.com/es/blog/back4app-com-es-las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/) Consultado el día 09 de enero de 2021.
- [8] <http://www.unam.mx/ptd/2015/08/08/Estadisticas-PUNAM-BUS-UNAM.pdf> Consultado el día 09 de enero de 2021.
- [9] <https://blog.back4app.com/wiki/Es-las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/> Consultado el día 09 de enero de 2021.
- [10] <https://datos.bancomundial.org/comercio/las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/> Consultado el día 09 de agosto de 2022.
- [11] [https://blog.back4app.com/es/las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/index.php?title=Microsoft\\_Visual\\_Studio&oldid=145587264](https://blog.back4app.com/es/las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/index.php?title=Microsoft_Visual_Studio&oldid=145587264)
- [14] <https://blog.back4app.com/es/las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/> Consultado el día 09 de diciembre de 2020.
- [12] Formadores(it). <http://www.formadoresit.es/xamarin-que-es-y-para-que-sirve/>

Consultado el día 15 de noviembre de 2020.



[13] back4app. <https://blog.back4app.com/es/las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/>.

Consultado el día 22 de octubre de 2020.

## Aapéndice A

Es importante señalar que para la realización del proyecto no se recurrió a ninguna metodología, sino más bien un método.

Para la elaboración de este proyecto primeramente se partió por investigar las diferentes plataformas para programar aplicaciones multiplataforma y analizar las características de cada una para al final tomar la mejor decisión. Una vez elegida la plataforma se hizo un bosquejo del diseño de la aplicación con todos los elementos que se deseaba tener. Seguido se procedió a la programación de la misma.

Comparación de plataformas de programación de aplicaciones multiplataforma.

### Back4App

“Back4App es una plataforma de desarrollo de aplicaciones de backend como servicio que permite a las empresas crear aplicaciones con gran eficiencia. Con esta solución de backend de bajo código, los desarrolladores solo necesitan enfocarse en los frontend de sus proyectos de aplicaciones porque Back4App asume la responsabilidad de las operaciones del lado del servidor.

### Características

- **Fácil integración:** ya sea que desee desarrollar una aplicación web o móvil, es fácil de integrar con Back4App. Esta plataforma de desarrollo de aplicaciones también catequiza y recopila datos relacionales a través de las API REST y GraphQL y varios SDK, incluidos iOS, Android, JavaScript, Flutter, etc.
- **Consulta en vivo:** esta función permite a los codificadores almacenar, consultar, eliminar y sincronizar los datos de la aplicación de forma espontánea.
- **Notificaciones por correo electrónico y push:** las aplicaciones avanzadas prefieren ir con las opciones de notificación por correo electrónico y push. En este sentido, las empresas pueden utilizar el panel de Back4App para enviar notificaciones automáticas y enviar correos electrónicos a sus usuarios.

- **CDN rápido:** la red de entrega de contenido de esta plataforma de desarrollo de aplicaciones permite a las empresas cargar y entregar sus archivos en cualquier momento y en cualquier lugar del mundo. Esta propiedad también garantiza el alojamiento y la entrega de archivos de forma rápida y segura.

## **Firestore**

Firestore es otra plataforma de desarrollo de aplicaciones que permite a las empresas crear aplicaciones web y móviles.

Como es una plataforma adquirida por Google, es por eso que los desarrolladores confían mucho en ella para construir backend de sus aplicaciones. Alrededor de 21,972 desarrolladores afirmaron en StackShare que usan Firestore.

Además, Instacart, Alibaba Travels y Twitch son gigantes tecnológicos populares que usan Firestore.

## **Características**

- **Bases de datos:** Firestore otorga dos tipos de bases de datos a sus usuarios. El primero es la base de datos en tiempo real. Esta es una base de datos NoSQL que permite a los usuarios sincronizar, recopilar y consultar datos en muy poco tiempo. La segunda base de datos es Cloud Firestore, que es básicamente un producto más avanzado de esta plataforma de desarrollo de aplicaciones. Cloud Firestore permite a los desarrolladores crear aplicaciones sin servidor con excelentes propiedades de seguridad de datos.
- **Configuración remota:** esta característica de Firestore permite a los programadores personalizar, probar y actualizar aplicaciones con control total.
- **Alojamiento seguro:** no importa desde dónde un desarrollador esté construyendo o iniciando una aplicación. Con Firestore, puede alojarlos de forma segura. En este sentido, Firestore proporciona back hosting SSD y certificados SSL gratuitos.

- **Almacenamiento en la nube:** con el respaldo de Google, es sólido para recopilar y entregar contenido mediante el almacenamiento en la nube. También es beneficioso en caso de que un usuario pierda la conexión a Internet. Una vez recuperada la conectividad, reanuda la carga o descarga desde el mismo punto donde la dejó el usuario”

**Fuente:** Presta, M. Las 10 mejores plataformas de desarrollo de aplicaciones. back4app. <https://blog.back4app.com/es/las-10-mejores-plataformas-de-desarrollo-de-aplicaciones/>.

## **Xamarin forms**

Xamarin es un software imprescindible para el desarrollo de aplicaciones móviles, puesto que permite a los desarrolladores compartir hasta un 90% del código entre plataformas distintas, utilizando un único lenguaje de programación, C#, y que cada vez está ganando más adeptos tanto por el rendimiento como por la posibilidad del acceso al API nativo.

Compilado de forma nativa, Xamarin es una buena herramienta para crear aplicaciones de alto rendimiento con aspecto nativo. Cuenta con Xamarin.iOS y Xamarin.Android como principales clientes, que compilan de forma automática el código fuente con el método Ahead-of-time en caso de iOS y con lenguaje intermedio y posteriormente en AOT para Android, con la posibilidad de adaptarse para solucionar posibles problemas como la asignación de memoria.

## **Características**

- **Enlaces fuertemente tipados:** Xamarin no solamente nos proporciona enlaces para prácticamente todos los SDK de plataforma subyacentes en Android o iOS, sino que también nos los presenta robustamente tipados, facilitando su uso y reduciendo el número de errores en tiempo de ejecución con unas aplicaciones de mayor calidad.
- **Amplia variedad de código de terceros:** Esta herramienta de programación de apps móviles dispone de una interoperabilidad con lenguajes Objective-C Java, C y C++, que nos da la opción de usar

bibliotecas existentes de Android o iOS escritas en dichos lenguajes, o enlazar bibliotecas nativas de Java o Objective-C.

- **Importantes mejoras con el lenguaje C#.** Gracias al uso de un lenguaje moderno, Xamarin ofrece construcciones funcionales como lambdas o características de programación en paralelo.
- **Compatibilidad con XML o bases de datos.** Xamarin es una herramienta que usa bibliotecas de clases base (BCL), ofreciendo nuevas características optimizadas, entre las cuales encontramos la compatibilidad con XML, bases de datos o redes.
- **Sistema de administración de proyectos y soluciones.** Al usar Visual Studio, un entorno de desarrollo integrado (IDE) moderno, Xamarin aporta nuevas características como un potente sistema de administración de proyectos y soluciones, así como la finalización automática de código o un mayor control del mismo.
- **Multiplataforma móvil.** Xamarin es compatible con las tres principales plataformas, compartiendo hasta un 90% del código en iOS, Android o Windows a través de una AP unificada de acceso a recursos comunes. De esta manera, se reducen drásticamente los costes de desarrollo y los plazos de comercialización para los desarrolladores de dispositivos móviles.

Fuente: Peña, A. (22 de junio 2020). Xamarin ¿Qué es y para qué sirve?. FORMADORES(IT). <http://www.formadoresit.es/xamarin-que-es-y-para-que-sirve/>

Estas fueron las principales plataformas que se tomaron en cuenta para el desarrollo del proyecto, después del análisis riguroso de las características, se llegó a la decisión de hacer el desarrollo con Xamarin forms. A diferencia de las otras plataformas Xamarin ofrece compartir hasta 90% de código.