



Universidad Nacional Autónoma de México



Facultad de Ingeniería

Ingeniería en Computación

Cómputo Móvil

Equipo:06

Tercer Parcial: App FI-Horario

Profesor: Ing Marduk Pérez de Lara Domínguez

Fecha de entrega: 06/01/2023

Semestre 2023-1

Integrantes:

- Avendaño Cabanillas Gustavo Eduardo
- Anizar Morales Victor
- Pérez Duarte Ana Patricia
- Torrecilla Jiménez Aarón Israel
- Villafañe Pérez Pamela Irais

ÍNDICE

Wireframes	3
Funcionalidades y gestos	4
Análisis de datos, requisitos del dispositivo y servicios	7
Prototipo/Maqueta de la app	8
Modelo de negocio y Estimaciones	8
Referencias	8

Wireframes

La aplicación contará con 5 pantallas, las cuales se muestran a continuación en la Figura 1, mismas que tendrán conexión entre sí, dependiendo lo que el usuario desee realizar:

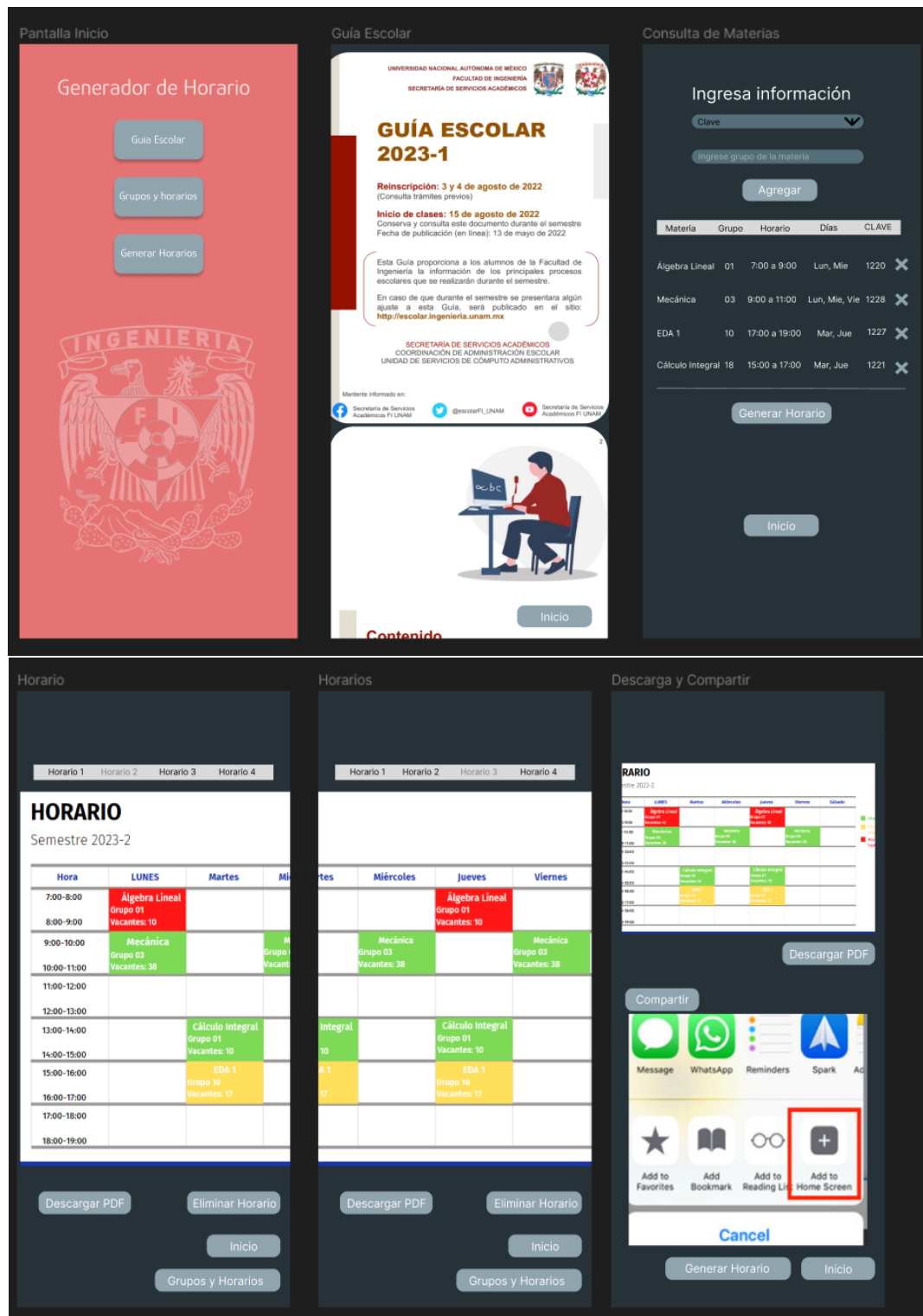


Figura 1. Propuesta de Pantallas.

Funcionalidades y gestos

Con fines más gráficos en el entendimiento del flujo de nuestra aplicación FI-Horario se creó el siguiente App-Flow, el cual se visualiza en la Imagen 2:

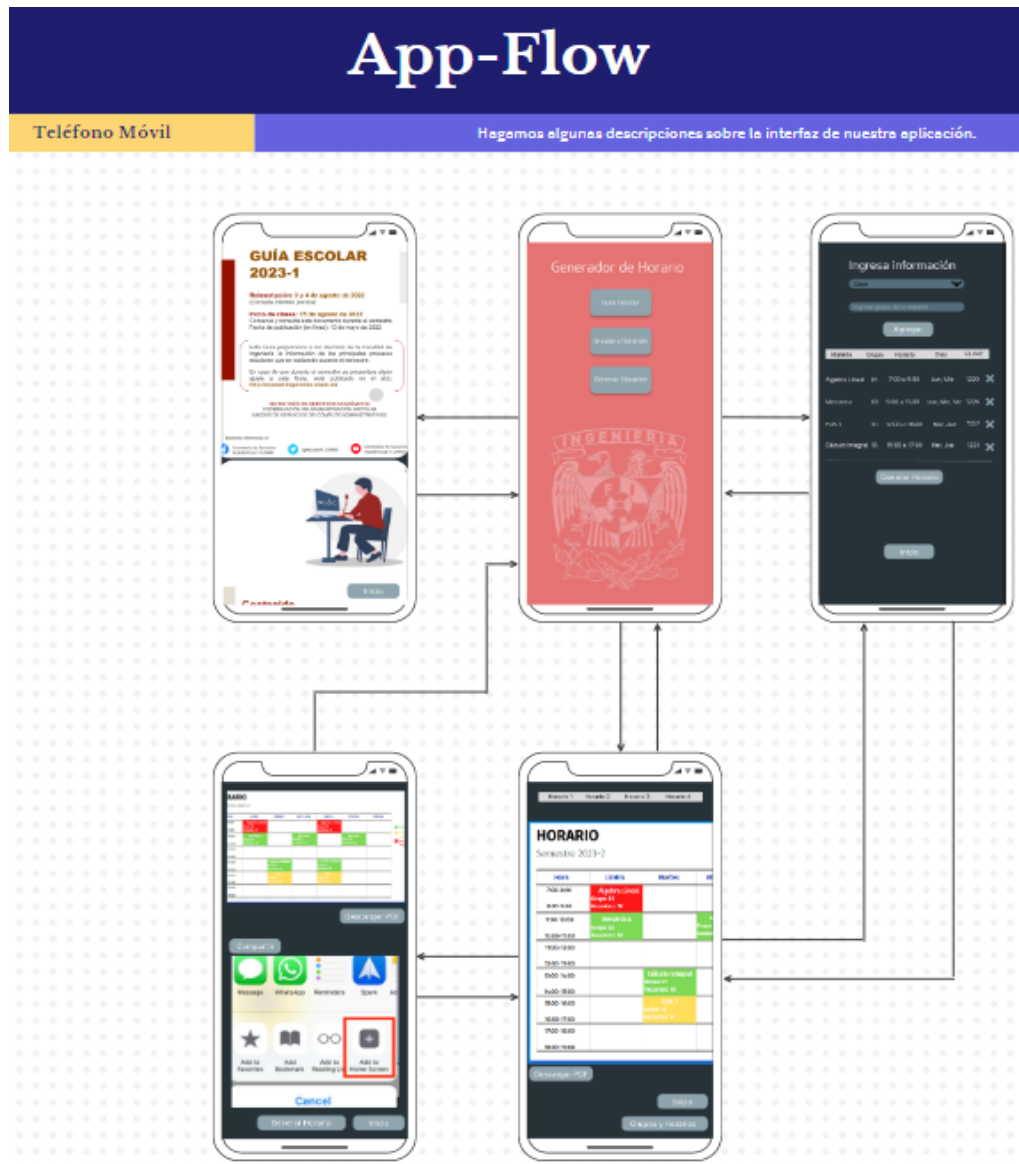


Imagen 2. App-Flow.

La pantalla principal que se encuentra en medio y de color rojo, tiene tres posibles opciones Guía Escolar, Grupos y Horarios y Generar Horarios, las cuales se describirán a detalle a continuación.

Guía Escolar: Para esta pantalla se tendrá acceso únicamente desde la pantalla de inicio, una vez ahí, solo podremos visualizar la guía escolar vigente, facilitando la información a la comunidad estudiantil, con la posibilidad de regresar a la pantalla principal mediante el botón de "Inicio".

Grupos y Horarios: Se tendrá acceso a esta pantalla desde el inicio, donde el usuario podrá dar de alta sus posibles materias a inscribir mediante la clave y el grupo deseado, una vez el usuario esté conforme con su decisión podrá dar click en el botón “Agregar”, el cual le mostrará un breve resumen de las mismas con la siguiente información: Materia, Grupo, Horario, Días y clave, de igual forma, se podrá regresar al inicio, y tendrá un flujo consecuente a la pantalla de Horario, que se describe a continuación.

Generar Horarios: La pantalla de horarios permitirá al usuario tener hasta 4 posibles opciones de horarios a visualizar, de acuerdo al horario elegido se podrá ver en detalle las vacantes disponibles de las materias y la opción de poder descargar el horario en formato PDF. Como página antecedente tendrá la de Grupos y Horarios y como consiguiente la de Compartir Horario.

Descargar y Compartir: En esta pantalla nos muestra una previsualización del documento en formato PDF y el botón para descargarlo, así mismo tenemos la opción para compartir documento a otras aplicaciones como Whatsapp, Telegram, etc y dependiendo del sistema operativo algunas opciones más. Como pantalla antecedente tendremos la de Generar Horarios y como consiguiente habrá dos caminos, la pantalla de inicio y la pantalla de Generar Horarios.

Como bien sabemos, la navegación dentro de nuestros teléfonos móviles ha evolucionado, desde la barra de navegación de los tres botones a los gestos especiales que implementa cada sistema operativo. Por ejemplo para Android deberemos activar estos gestos mediante las configuraciones del sistema, una vez activados sólo bastará con deslizar el dedo hacia el lado izquierdo para regresar a la actividad anterior, realizar captura de pantalla con deslizar tres dedos hacia abajo etc, y pasa exactamente lo mismo con Apple.

En general, para nuestra aplicación se usarán los gestos por default del dispositivo móvil que el usuario posea, ya sea Android o Apple que se resumen de la siguiente manera:

- Deslizar a la izquierda. Realizará un retroceso de pantalla dentro y fuera de la aplicación.

- Deslizar a la derecha. Se podrá avanzar a las pantallas siguientes definidas por la programación de la app.
- Deslizar de abajo hacia arriba. El usuario podrá volver a la pantalla principal de su dispositivo.
- Deslizar hacia arriba y mantener por unos segundos. Se podrá ver las aplicaciones abiertas en segundo plano.
- Barra luminosa en la parte inferior. Si la deslizas hacia la derecha o izquierda podrás cambiar de aplicaciones rápidamente.
- Zoom in y zoom out. Dentro de la pantalla de Horario, Guia Escolar y cuando el usuario decida tener su posible horario en un formato pdf se realizará de la siguiente manera:



Pngtree. (s. f.). *Hand Gesture Zoom In Zoom Out Vector and PNG*.
https://pngtree.com/freepng/hand-gesture-zoom-in-zoom-out_5373683.html

De esta manera podremos mejorar la visualización del horario haciendo un “zoom in” pudiendo acercarnos a la sección del horario que deseemos ver con detalle.



Pngtree. (s. f.). *Hand Gesture Zoom In Zoom Out Vector and PNG*.
https://pngtree.com/freepng/hand-gesture-zoom-in-zoom-out_5373683.html

Mediante este gesto, haremos un “zoom out” que nos regresará a la escala inicial de visualización, incluso permitiéndonos alejarnos más de la escala inicial fijada por la app por defecto.

- Mantener presionado. Otro gesto que se implementará en la aplicación será dentro de la pantalla de Horario, donde, en el caso que la materia que desee el usuario en inscribir ya no tenga vacantes o esté cerca de quedarse sin vacantes manteniendo presionado el recuadro asignado a la materia, tendrá la opción de cambiar la materia a otro grupo, mismo que se actualizará en la misma pantalla o simplemente eliminarla y tener la opción de darla de alta en la pantalla de Alta de materias.

Análisis de datos, requisitos del dispositivo y servicios

Nuestra aplicación web está desarrollada para cualquier tipo de dispositivo móvil que pueda navegar en Internet a través de un navegador, el almacenamiento que tenga el usuario no será de interés ya que será un espacio mínimo comparado con otras aplicaciones móviles actuales, los datos con lo que se trabaja se almacenan y procesan en la web, por lo que únicamente será necesaria la conexión a una red. Aunado a lo anterior son aplicaciones compatibles y accesibles (ejecución en distintos sistemas operativos y accesibles desde cualquier dispositivo); y el mantenimiento y la actualización es sencilla (misma versión para todos los usuarios).

Podemos concluir que la App FI-Horario está desarrollada para cualquier dispositivo que cuente con conexión a internet y sea capaz de ingresar mediante un navegador, por ende los tamaños de pantalla pueden variar de acuerdo al dispositivo en cuestión, en cuanto a la orientación, se plantea de forma inicial sea únicamente vertical, en caso de que los usuarios necesiten visualizar más acerca de su horario, se tendrá la posibilidad de cambiar la vista a forma horizontal.

Con el fin de ser una aplicación de uso general, no se requiere usar sensores específicos del dispositivo móvil del usuario, únicamente será necesario que funcione correctamente la pantalla táctil del usuario para que tenga una experiencia óptima y logre interactuar, visualizar y pueda hacer uso de la app tal y como se tiene planteada sin ningún tipo de impedimento.

Hablando de los servicios y tipos de datos que utiliza nuestra aplicación se menciona enseguida por cada una de las pantallas.

Pantalla de inicio: La pantalla de Inicio no utiliza ningún servicio, únicamente contamos con botones que nos enlazan a las otras pantallas de la app.

Guia escolar: Para esta pantalla se utiliza un endpoint de nuestra Api (realizada con fast Api), el nombre de dicho endpoint será “GetGuiaEscolar”. Cómo se menciona en el mismo nombre del endpoint estamos hablando de un método get, el cuál nos devuelve un tipo de dato binario que contendrá el PDF de la guía escolar, esta guía será actualizada semestre tras semestre y se almacenará dentro del servidor que utilizaremos.

Grupos y Horarios: En esta pantalla se hará uso de un endpoint nombrado “PostDatosGrupo”, este endpoint recibirá como parámetros la clave de la materia a dar de alta y el número de grupo. En este caso estamos hablando de un método de tipo post. Internamente la Api hará una consulta en la página de horarios de la facultad de ingeniería y el scraper hará el trabajo de realizar un análisis de los datos obtenidos de tal manera que en la respuesta obtengamos un objeto como el siguiente:

```
Grupo: {  
  
    Materia:    string,  
    Grupo:      int,  
    Horario:    string,  
    Dias:       string,  
    Clave:      string  
  
}
```

* Objeto de tipo Grupo.

Para obtener esta respuesta es necesario que la página de horarios de la facultad de ingeniería se encuentre trabajando con normalidad, ya que el scraper depende de los datos que obtiene de esta página, de lo contrario el endpoint nos regresará un error con la siguiente descripción: “grupo o materia no encontrado”

Una vez obtenido estos datos en el frontend se agrega un registro en la tabla de Grupos y Horarios.

Generar Horarios: Esta pantalla utiliza un método Post la cual recibe una lista de Objetos de tipo Grupo, el método fue nombrado "PostGenerarHorarios". Internamente este método realiza un análisis de la lista de Grupos añadidos. El método regresa una lista con datos de tipo binario que son los pdfs generados, con opciones adicionales. Si deseas descargar o compartir simplemente con el botón correspondiente vas a la pantalla "Descargar o compartir" y se le pasa el el dato binario a la pantalla.

Descargar y compartir : Al entrar a esta pantalla ya contamos con el dato binario (PDF) que describe el horario. Estos datos binarios son brindados por la pantalla anterior.

La api no almacena datos en ningún momento ya que todo el procedimiento consiste en realizar consultas y analizar los resultados a través de scraping data en tiempo real. El costo de esta api es verdaderamente bajo ya que existe mucha documentación y ejemplos de cómo realizar un scraper, por lo que no se necesitará de un programador verdaderamente experimentado para realizar esta tarea.

Prototipo/Maqueta de la app

El siguiente link direcciona a un video con el demo de la aplicación:

- https://youtu.be/_UcrYCuJNTA

Modelo de negocio y Estimaciones

Las herramientas con las cuales nuestra aplicación será desarrollada son varias, llegamos a la conclusión de que será una aplicación híbrida. Por un lado tendremos a Flutter con el cual se estará desarrollando el frontend, todo el aspecto gráfico o visual de nuestra aplicación. Dicho lo anterior podemos decir que el lenguaje de programación para el frontend será Dart.

En cuanto al backend será desarrollado con el lenguaje de programación Python, más en específico se usará al framework FastApi. En cuanto a motor de base de datos se usará a MongoDB principalmente debido a todos los beneficios y facilidades que nos ofrece trabajar con una base de datos no relacional.

El principal motor por el cual le estaremos ofreciendo al usuario final la información es por medio de la estrategia "Scraping Data", que, a grandes rasgos, es obtener

información específica a partir de una dirección web en concreto. Es muy importante mencionar que esta estrategia de información solo será en etapas tempranas de la aplicación, eventualmente se buscará llegar a acuerdos/negocios con las respectivas organizaciones y/o directivos de la facultad para que la información principal de nuestra aplicación sea nativa en nuestro sistema.

En esta aplicación se tendrán distintos roles de desarrollo de software: Uno o varios desarrolladores Frontend y Backend, así como también un arquitecto de bases de datos; este último será opcional en un principio dado que en etapas tempranas de la aplicación, el manejo de los datos será simple dado que se usará el Scraping, para este aspecto el desarrollador backend podría encargarse de ello. Sin embargo, conforme vaya creciendo la aplicación así como sus datos, será necesario un rol que se encargue específicamente de esto. Es importante mencionar que no nos cerramos a la posibilidad de trabajar únicamente con uno o varios desarrolladores Full Stack, sobre todo en las primeras etapas de la aplicación. Conforme crezca la aplicación, es muy probable que se vayan necesitando roles más específicos para el desarrollo de la misma, por ejemplo, puede que en un futuro se implementen funcionalidades relacionadas a la inteligencia artificial: en esos posibles casos ya se requerirían puestos más específicos dentro del área de desarrollo de software. Los desarrolladores serán los encargados de toda la parte técnica de la aplicación, es decir, son ellos quienes estarán completamente inmersos en el código fuente, las bases de datos, la gestión de los datos (En un principio se usa Scraping, pero eventualmente se pueden requerir API 's).

Aparte del equipo 100% técnico se requiere de uno o dos Project manager para que puedan gestionar en su totalidad el desarrollo del proyecto, este rol será quien describa la mejor forma de trabajar de los programadores. Por ejemplo, puede decidir si trabajamos con una metodología en específico. Será quién tenga un contacto más cercano con el usuario final; levantará los requerimientos de los usuarios y estará en contacto con el resto del equipo en caso de que se requiera algo.

En cuanto a tiempos de desarrollo y sus costos: Primero se tiene que hacer una planeación inicial en cuanto a las actividades a seguir (2 días), posteriormente se tiene que hacer una serie de reuniones con el equipo de trabajo para analizar la

situación actual por la cual también se está creando la aplicación (1 semana), posteriormente se empieza con toda la parte técnica: creación de base de datos junto con su configuración (2 días), creación de la aplicación como tal (8 semanas). Este último punto es considerando la creación de backend (3 semanas) y frontend (5 semanas). Posteriormente se tiene que hacer un análisis previo a la entrega final y eventual publicación de la aplicación (1 semana). Dicho lo anterior para esta primera etapa se considera un tiempo estimado de 10 a 12 semanas. Sin embargo, es importante mencionar que una vez publicada la aplicación, se estará haciendo un trabajo continuo para seguir ofreciendo mejoras o corrigiendo errores; este mantenimiento es de suma importancia y será un lazo muy fuerte con la comunidad estudiantil. Se tiene planeado sacar una actualización a la aplicación cada cierto tiempo o, en casos extraordinarios, cuando sea sumamente necesario.

Unidad de tiempo Días.						
Columna1	Columna2	Columna3	Columna4	Columna5	Columna6	Columna7
Tarea/Actividad	Descripción	Optimista(O)	Pesimista(P)	Más probable(m)	Distribución Triangular	Distribución Beta
Planeación inicial	Toma de Requerimientos Funcionales	1	4	2	2.333333333	2.166666667
Reuniones	Reuniones ...	3	8	5	5.333333333	5.166666667
Creación de Base de Datos	Analizar y generar la base de datos a utilizar.	1	4	2	2.333333333	2.166666667
Desarrollo		35	48	40	41	40.5
Backend	Programación en lenguajes HTML, Python e implementación de framework	12	25	15	17.33333333	16.16666667
Frontend	Diseño, programación e implementación de las pantallas y navegación.	23	35	25	27.66666667	26.33333333
Análisis previo a entrega	Bosquejo general del funcionamiento de la app.	3	15	5	7.666666667	6.333333333
Pruebas	Realizar pruebas de uso para solucionar posibles fallos o bugs dentro de la aplicación.	3	10	5	6	5.5
Entrega	Entrega al cliente.	1	3	1	1.666666667	1.333333333
Total (días)					111.3333333	105.6666667

Imagen 3. Estimación de tiempos.

De acuerdo a los cálculos realizados y apoyándonos en la distribución “Beta”, que es un método de cálculo mucho más aproximado entre todos los parámetros, podemos concluir que tendremos una duración estimada de 106 días para el desarrollo del proyecto.

Estimaciones de costos.

Unidad de tiempo Días.						
Columna1	Columna2	Columna23	Columna22	Columna225	Columna224	Columna223
Tarea/Actividad	Descripción	Esfuerzo(Horas)	RecursosHombre	RecursosEquipos	CUnitarioOptimista	CUnitarioPesimista
Planeación inicial	Toma de Requerimientos Funcionales	48	5	1	15	10
Reuniones	Reuniones ...	120	5	1		
Creación de Base de Datos	Analizar y generar la base de datos a utilizar.	48	2	2	70	40
Desarrollo		960	3	3		
Backend	Programación en lenguajes HTML, Python e implementación de framework	360	2	2	18	12
Frontend	Diseño, programación e implementación de las pantallas y navegación.	600	1	1	12	8
Análisis previo a entrega	Bosquejo general del funcionamiento de la app.	120	5	5		
Pruebas	Realizar pruebas de uso para solucionar posibles fallos o bugs dentro de la aplicación.	120	1	1	14	9
Entrega	Entrega al cliente.	24	2			

Imagen 4. Estimación de costos.

Columna222	Columna3	Columna4	Columna5	Columna6	Columna7
CUnitarioMasProbable	COptimista(CO)	CPesimista(CP)	CMás probable(Cm)	Distribución Triangular	Distribución Beta
12	3600	2400	2880	2960	2920
62	6720	1920	5952	4864	5408
16	12960	8640	11520	11040	11280
10	7200	4800	6000	6000	6000
12	1680	1080	1440	1400	1420
Total en pesos				26264	27028

Imagen 5. Estimación de tiempos (continuación).

De acuerdo a las figuras presentadas previamente y mediante el cálculo de los costos unitarios (calculados en pesos MXN por hora), los recursos humanos y los distintos parámetros, tenemos calculado tomando en cuenta la distribución Beta, un costo de 27,028 MXN para concretar el desarrollo del proyecto, pagando a todo el equipo de acuerdo a sus respectivos roles en el desarrollo de la app móvil.

Referencias

1. Pngtree. (s. f.). *Hand Gesture Zoom In Zoom Out Vector and PNG*.
https://pngtree.com/freepng/hand-gesture-zoom-in-zoom-out_5373683.html
2. Caballero, D. (2022, 28 junio). *Controla tu móvil Android o iPhone con los gestos de su pantalla*. MovilZona.
<https://www.movilzona.es/noticias/power-on/controla-android-gestos/>
3. Maluenda, R. (2022, 5 mayo). *Tipos de desarrollo de aplicaciones web: ejemplos y características*. Profile Software Services.
<https://profile.es/blog/desarrollo-aplicaciones-web/>
4. Just a moment. . . (s. f.).
https://www.canva.com/design/DAFV0_JX8wQ/j4uJjBAYBDZLwLg3-4ggyQ/edit?utm_content=DAFV0_JX8wQ