



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Asignatura: Cómputo Móvil

Grupo: 2

Semestre: 2022-2

**Examen 3- Especificaciones técnicas de la app
“Cornerworker”**

Fecha Límite de Entrega: 31/05/2022

Profesor: Marduk Pérez de Lara Dominguez

Alumnos:

- **Najera Noyola Karla Andrea**
- **Facundo Acuña Edgar**
- **Cruz Galván Alberto Israel**

Especificaciones de la app “Cornerworker”

NOTA: Debido al tamaño del contenido del Wireframe, se añade toda la información respectiva al final del documento.

- **Especificaciones técnicas (dispositivos destino, almacenamiento, gestos)**

La aplicación estará diseñada de manera prácticamente exclusiva para celulares, contando con un diseño responsivo para adaptar el tamaño a las distintas resoluciones del mercado. Se pretende trabajar la interfaz exclusivamente en un formato Portrait (retrato), por lo cual no se permitirá que el usuario gire su dispositivo para cambiar de orientación. Cabe destacar que, en el caso de Android, la versión mínima requerida será la 5.0 y la máxima será el último sistema operativo que fue lanzado para celulares (actualmente Android 12). Para iOS solo llegará a las versiones que están entre iOS 13 y iOS 15 debido a las restricciones de AppStore por conservar apps nuevas en dispositivos novedosos.



Dado que nuestra aplicación será un portal de búsqueda/oferta de trabajos correspondientes al sector informal, no planeamos utilizar un gesto o movimiento propio en el que intervenga el giroscopio o la misma pantalla del dispositivo. Solo se busca que el usuario recorra la aplicación en un modelo jerárquico plano, con los gestos tradicionales de deslizamiento vertical, horizontal y lateral. Y el único sensor que requerimos es el de la cámara y el GPS para los casos que convengan (foto del INE o de los servicios, además de la localización de la persona). Debido a lo anterior, se solicitaron los permisos de uso de cámara y ubicación (idealmente el usuario debe aceptarlos para usarlos solo cuando la app está en uso).

Asimismo, el método de almacenamiento será completamente en la nube, por lo que se requiere que en todo momento se tenga una conexión a internet que, de no ser así, simplemente se desplegará un mensaje avisando al usuario que no puede acceder a la app y que se requiere de una conexión a internet. No obstante, será necesario que se pueda almacenar internamente (en caché) el correo y contraseña para acceder a la cuenta con el fin de tener la cuenta abierta el mayor tiempo posible y acceder a ella cuando sea necesario sin necesidad de iniciar sesión en cada ocasión.



- **Análisis de los servicios de back que necesitan conectar o implementar. Detallando qué datos fluyen, si son de consulta, registro, borrado o actualización. También si ya existe el servicio detallar las características, costos y requerimientos para conectarse a él (como medios de autenticación, formatos, etc)**

Nuestra aplicación funcionará con servicios de la nube Azure, con dicho servicio pagaremos sólo por lo que consumamos y permitirá escalar nuestros proyectos según lo vayamos necesitando.



Elegimos como proveedor de nube a Azure porque nos permitirá implementar y escalar nuestra aplicación. Además, este entorno de infraestructura es adecuado si queremos integrar tecnologías tales como:

- .NET
- Node JS
- Java
- Python
- PHP

Tanto para la aplicación en el OS IOS como Android, será necesario implementar un CRUD a nuestra app para que se pueda dar de alta y baja de usuarios, así como también para toda la lógica de contratación de servicios o al darse de alta los usuarios de nuestra app.

Creemos que lo más conveniente es utilizar una arquitectura de microservicios, para nuestra aplicación, esto en lugar de utilizar una arquitectura monolítica.

Las ventajas de utilizar este tipo de arquitectura son las siguientes:

- Mayor escalabilidad
- Posibilidad de integrar más tecnologías y lenguajes de programación
- Al segmentar la funcionalidad de nuestra app por microservicios, podremos tener una alta disponibilidad y en el caso en el que un microservicio falle, sólo será

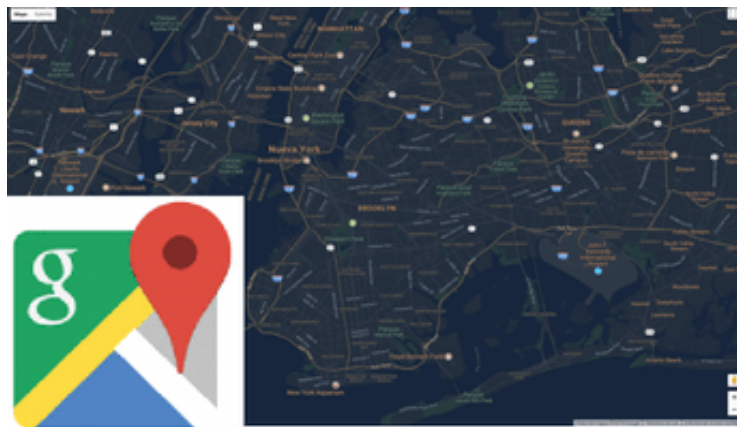
cuestión de atender dicha incidencia, puesto que las demás funcionalidades de la app seguirán ejecutándose y atendiendo las solicitudes de los usuarios.

Las posibles desventajas de utilizar este tipo de arquitectura son:

- De no contar con un diseño adecuado, la complejidad de nuestra aplicación aumentará.

Para la aplicación *CornerWorker* es un requerimiento funcional agregar una API de mapa de servicios para poder conectar a nuestros usuarios. La elección fue utilizar Google Maps Platform, que es un conjunto de API y SDK que los desarrolladores pueden usar a fin de incorporar Google Maps en apps para dispositivos móviles y páginas web o recuperar datos de Google Maps. Se ofrecen diferentes productos. Se hará uso de los SDK que cuentan los servicios de Google Maps Platform para Android y iOS, también se puede ofrecer información adicional sobre las ubicaciones del mapa y facilitar la interacción con el usuario si se agregan marcadores, polígonos y superposiciones en el mapa.

El SDK es compatible con los lenguajes de programación Kotlin y Java para Android, y Swift para iOS. Además ofrece bibliotecas y extensiones adicionales para funciones avanzadas y técnicas de programación. El método de cobro de las API de Google Maps Platform se facturan según el SKU, se realiza un seguimiento del uso de cada SKU de producto, y una API puede tener más de un SKU de producto. SKU: Mobile Native Dynamic Maps, SKU: Dynamic Street View, SKU: Dynamic Maps.



También necesitaremos usar aplicaciones externas como el teléfono, el administrador de archivos y la cámara para los cuales no se necesita una API como tal, sino que el usuario brinde el acceso a la app a estos servicios y, con ello, garantizar un adecuado funcionamiento de la app.

- **Detalles sobre el o los lenguajes de programación que usarán y las herramientas para desarrollar.**

A continuación, se desarrollará la información detallada de las principales tecnologías a utilizar en nuestra app:

Herramienta - Lenguaje - Tecnología	Descripción	Frontend	Backend	Otro servicio y cuál
Git / GitLab	Herramienta para control de versiones			Versionamiento de código
Swift	Lenguaje para desarrollo iOS			
Xcode	IDE desarrollo para swift			Entorno de desarrollo
Kotlin	Lenguaje de programación para Android			
Selenium	Herramienta de pruebas			Automatización de pruebas
PgAdmin	Manejador de bases de datos			
Data modeler	Modelador de datos			Modelado de bases de datos
Whimiscal	Diseño de wireframes y diagramas en general			Diseño
Drawio	Diseño de diagramas en general			Diseño
Java - Spring	Para integración de microservicios			
.NET	Para integración de microservicios			
Python - Django	Para integración de microservicios			
IntelliJ idea	IDE de desarrollo para java			Entorno de desarrollo
Visual Studio Code	Editor de código			Editor de código

- **Equipo de trabajo y roles que intervienen para realizarla.**

El equipo de trabajo se establecerá de acuerdo a la metodología de trabajo Scrum, Scrum es un proceso de ingeniería de software iterativa para desarrollar y entregar software.

El marco Scrum está comprendido por 3 roles.

- Product Owner, representa a los clientes finales y otras partes interesadas, y es el responsable de maximizar el valor del producto, asegurando que el equipo scrum entregue su trabajo en el tiempo correcto, además, realiza el levantamiento de requerimientos, gestiona y aclara los requisitos del proyecto.
- Los miembros del Team Scrum son los que implementan el software. Estos decidirán conjuntamente la cantidad de requisitos o requerimientos que se pueden cumplir durante un incremento de producto en particular llamado Sprint.

- Scrum Master para dirigir el proyecto y proteger tanto los intereses del cliente como los del empleado y de la empresa. Además, se encarga de ejecutar y aplicar correctamente el trabajo del Marco Scrum.

Profundizando en los roles del proyecto e integrando las partes interesadas.

- Analista. Se encargará de recabar información relevante para generar valor en la aplicación *CornerWorker*.
- Desarrolladores. Implementarán los recursos humanos para la creación del software.
- Testing. Se necesita la colaboración de personal para comprobar las funcionalidades del producto final así mismo de automatizar el envío de errores que tengan los usuarios.
- Director de Marketing. Será el encargado de interpretar los datos obtenidos del analista para generar una campaña publicitaria y crear una estrategia de lanzamiento de la app *CornerWorker*.
- Diseñador. Se encargará de dar forma y colores al producto generado por los desarrolladores. Así mismo de crear una interfaz amigable para los clientes y usuarios de la app *CornerWorker*.
- Analista Financiero. Su trabajo consistirá en evaluar los precios de mercado para poder generar una estrategia socio consumidor.
- Equipo legal: Serán los encargados de redactar el aviso de confidencialidad, así como términos y condiciones. Asimismo, revisarán las identificaciones de los socios para garantizar que son personas reales.

- **Estimaciones de tiempo de desarrollo y costos**

La estimación del tiempo de desarrollo es de 6 meses para la primera versión de *CornerWorker* esto contempla a un solo team Scrum de 5 desarrolladores siendo un monto de \$1,000,000.00 MXN que cubra este rubro.

Asimismo, se consideran dos especialistas en Scrum Master y product owner para dirigir y coordinar el proyecto de la aplicación siendo un monto de \$260,000.00 MXN.

También se ha considerado un presupuesto para personal poco involucrado en el desarrollo como un especialista en testing de software que validara los estándares de calidad, así

mismo de un diseñador, un analista financiero, director de marketing, un analista y 3 miembros de equipo legal. Estos roles cubrirán un recurso neto de \$700,000.00 MXN.

Por último se tienen un presupuesto para cubrir los insumos y recursos materiales que utilizarán los recursos humanos, considerando una cantidad de \$700,000.00 MXN.

De acuerdo a la visión del producto final y las estimaciones a largo plazo, se ha considerado que la inversión inicial de acuerdo a los campos propuestos anteriormente es de \$2,200,000.00 MXN.

- **Levantamiento de requerimientos**

ID RN	Regla de negocio	ID REQ	Requerimiento	Tipo de requerimiento
1	Para poder hacer uso de los servicios proporcionados por <i>CornerWorker</i> es necesario que los usuarios estén registrados en la plataforma.	1	Se debe contar con una pantalla de registro.	Funcional
2	Cualquier cliente debe contar y proporcionar un correo electrónico.	2	Debe contar con una validación de correo. Solo se admiten valores alfanuméricos y el símbolo de @.	Funcional
		3	No deben existir espacios en ninguna parte de la cadena de texto que compone el correo.	Funcional
3	Para proteger los datos de los clientes y proporcionar seguridad, deben contar con una contraseña.	4	Alfanumérico con caracteres especiales.	Funcional
		5	De 8 a 16 caracteres.	Funcional
4	El cliente debe contar con un número telefónico donde se puedan comunicar para saber de sus servicios.	6	Se debe contar con la validación del número telefónico de 10 números.	Funcional

5	El cliente deberá seleccionar su perfil de usuario.	7	Se tendrá un apartado de socios visible exclusivamente para las personas con este perfil. Para contratar la membresía, se debe hacer desde la creación de cuenta o en un momento posterior.	Funcional
		8	Si el usuario es proveedor de servicios, deberá proporcionar el o los oficios que ofrece a la comunidad.	Funcional
6	Para adquirir los servicios los clientes deben conocer los términos legales y aceptar la política de la empresa.	9	Se deberá disponer de un selector con dos opciones que permita al cliente aceptar los "Términos y condiciones".	Funcional
7	Se necesitan almacenar los datos personales de los clientes nuevos en una BD	10	Se necesita un botón que permite enviar y guardar el formulario en una BD.	Funcional
		11	En caso de ser socio, se pedirá y revisará la credencial de elector	Funcional
8	Para poder hacer uso de los servicios proporcionados por <i>CornerWorker</i> es necesario iniciar sesión.	12	Se debe contar con una pantalla de inicio de sesión.	Funcional
9	El usuario debe contar con un correo y contraseña válidos.	13	Se necesita un botón que corrobore la información ingresada por el cliente en la BD.	Funcional.
		14	Se debe mantener la sesión	Funcional

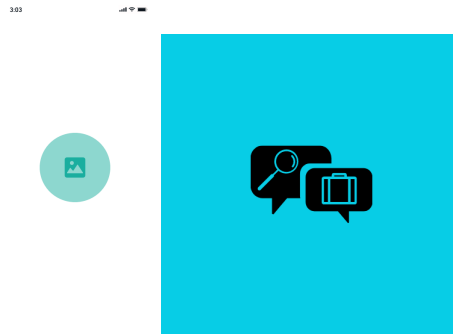
			abierta.	
10	Los usuarios podrán visualizar la información de los socios proveedores de servicios.	15	Se podrá visualizar los datos del socio que proporciona el servicio.	Funcional
		16	Debe ser visible la calificación del socio.	No funcional
11	La comunidad tiene la oportunidad de evaluar los servicios adquiridos por los socios de <i>CornerWorker</i> .	17	Se tendrá que haber adquirido y concluido un servicio para poder calificar al socio por medio de estrellas.	No funcional
		18	Se desplegará una pestaña que muestre la reseña (con estrellas) del socio evaluado.	No funcional
12	El usuario podrá hacer uso de un mapa que muestre los diferentes oficios que se proporcionan en su zona.	19	Una vez que el usuario seleccione un oficio se desplegarán los diferentes socios que puedan proveer el servicio requerido por el cliente.	Funcional
13	El usuario podrá visualizar y cotizar precios entre los diferentes socios.	20	Al seleccionar un socio con el oficio que pueda cumplir con el servicio, le aparecerán la información de los socios, indicando costos.	No funcional
		21	Si no se encuentra una opción adecuada, el cliente puede crear su propia solicitud.	Funcional
14	Para poder proporcionar servicios como socios por <i>CornerWorker</i> es necesario iniciar sesión en la plataforma.	22	Debe haber una pestaña que permita el cambio de usuario a socio.	Funcional.
15	El socio podrá agregar a su	23	Debe existir un apartado para	Funcional.

	perfil un catálogo de oficios que puede proveer.		agregar servicios	
16	El socio podrá agregar una lista de las actividades que puede ejercer con su oficio.	24	Se podrán agregar ofertas de servicio que sea capaz de ofrecer.	No funcional
		25	Se cuenta con la evaluación y trabajos anteriores de la persona (solo categoría y descripción).	No funcional
17	El socio podrá editar y eliminar la lista de las actividades que puede ejercer con su oficio.	26	Se contará con botones para editar y eliminar las ofertas.	Funcional
18	Se deberá indicar la disponibilidad de un socio para que pueda estar disponible para aceptar servicios.	27	Se deberá contar con un apartado donde se pueda desactivar las ofertas de servicios.	Funcional
19	El socio recibirá los servicios solicitados.	28	El socio podrá aceptar las solicitudes de servicios.	Funcional
		29	El socio sólo podrá aceptar trabajos si no tiene repetidas calificaciones muy bajas.	Funcional
20	El socio podrá cancelar las órdenes de servicio	30	El socio contará con un botón para cancelar los servicios solicitados.	Funcional
21	Los socios podrán visualizar la información de los usuarios.	31	Cuando los usuarios soliciten algún servicio podrán ver sus datos de contacto.	Funcional.
22	Los socios podrán calificar a los usuarios	32	Los socios podrán calificar el trato por parte de los usuarios	No funcional

- **Explicación del flujo y funcionalidades de las pantallas**

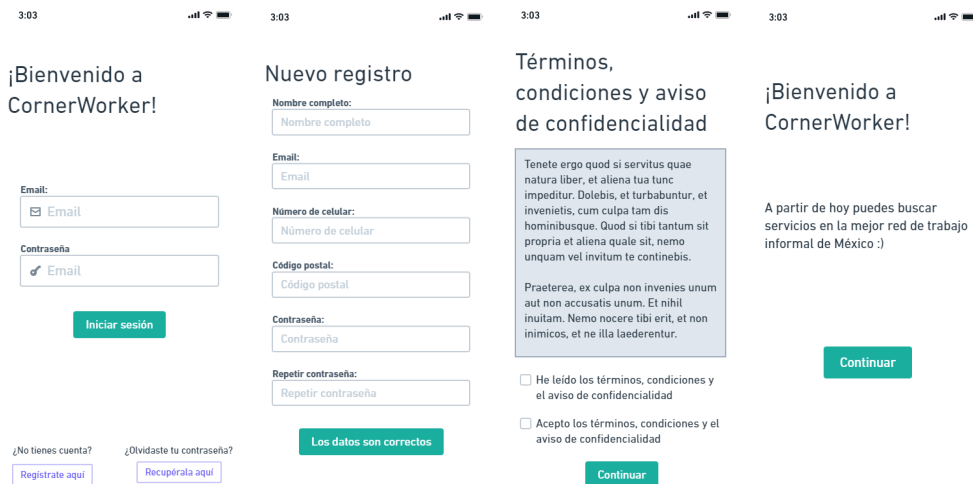
Para comprender el funcionamiento de nuestra aplicación y las pantallas que se utilizan para cada apartado, procederemos a mostrar cada una de las funciones que es capaz de realizar CornerWorker:

1. SplashScreen



Cada que se inicie la aplicación, se mostrará una splash screen con el logo de la aplicación. Cabe destacar que puede haber un pequeño error en dispositivos con Android 12 debido a la última actualización donde, por defecto, el sistema muestra en ocasiones dos veces dicha pantalla (una con el logo de la app y otra con el splash correspondiente).

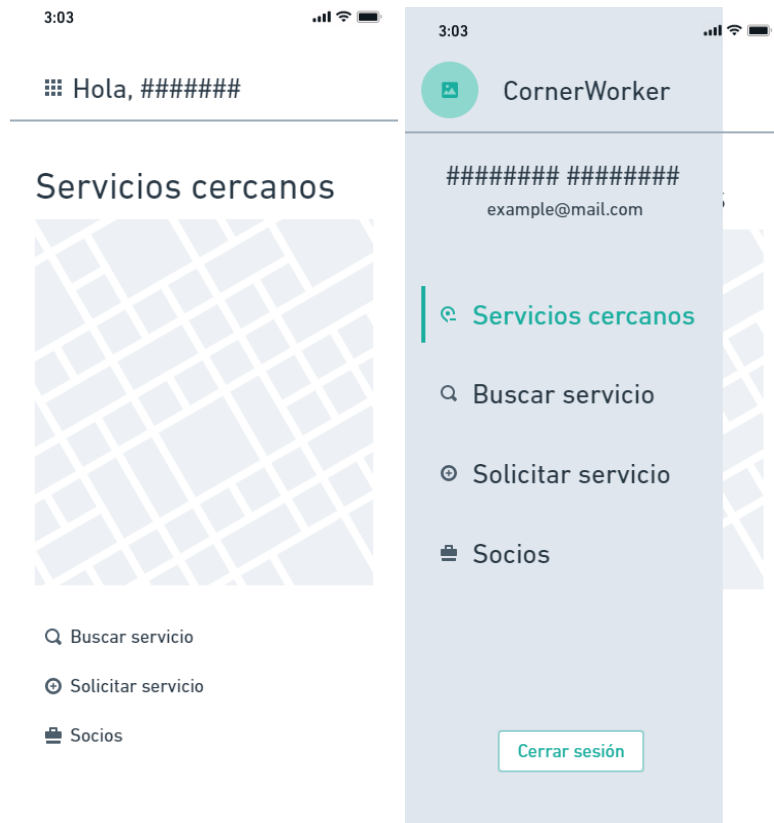
2. Registro y Login de usuarios



Tras iniciar la aplicación, se mostrará el apartado de Login del usuario. En caso de contar con una cuenta existente, el usuario simplemente ingresará y podrá acceder al menú de inicio de CornerWorker. No obstante, de no tener cuenta, se le presenta un formulario en el cual se le solicita nombre, correo electrónico, email y contraseña para la cuenta. Asimismo,

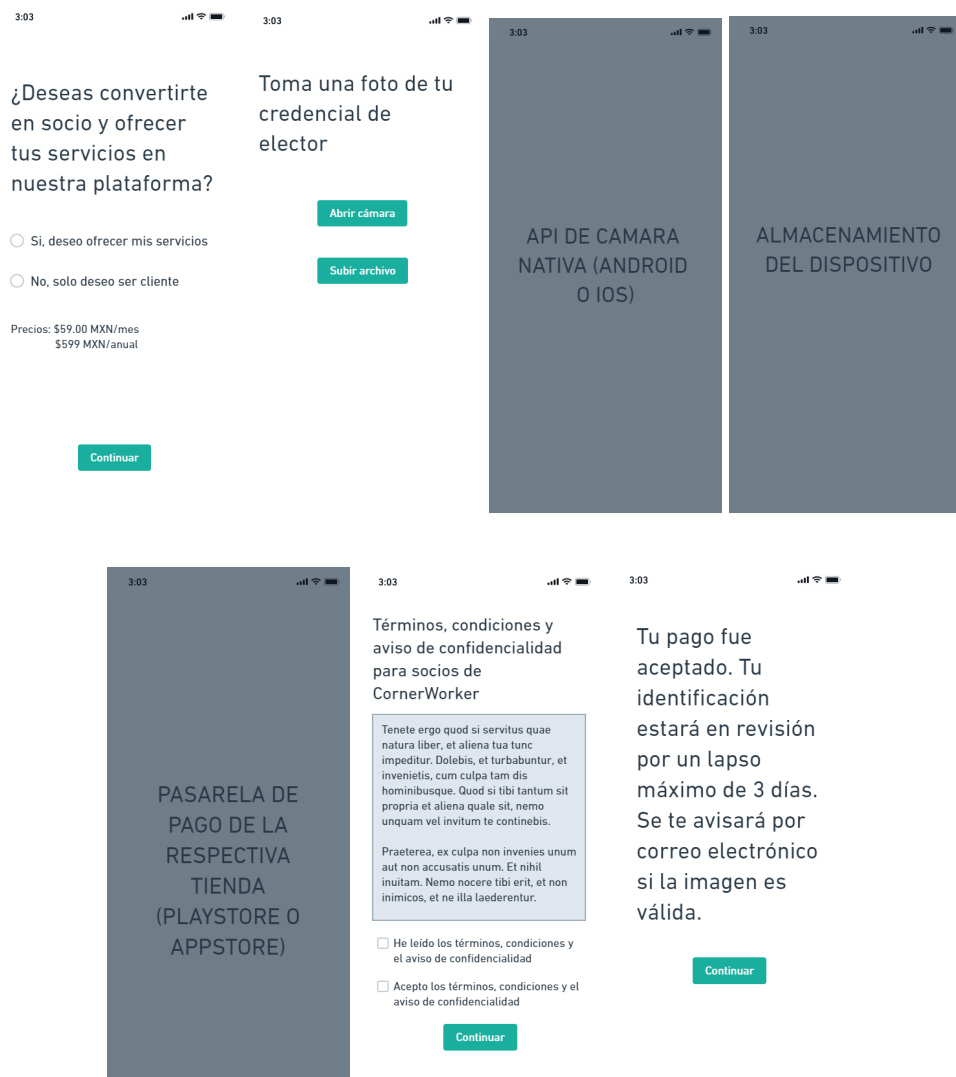
se le hace leer al usuario un apartado de términos, condiciones y aviso de confidencialidad tras lo cual debe aceptar estar de acuerdo. Tras aceptarlo, puede convertirse en socio de una vez (Para mayor detalle, revisar la función correspondiente más adelante). Una vez concluido el registro, se pasa al menú de inicio de la app.

3. Menú del usuario cliente



Una vez que se está dentro de la aplicación, se despliega un mapa en el que aparecerán los servicios que ofrecen diversos socios en la zona cercana al domicilio. En la barra superior aparecerá el nombre de la persona y un botón que activa el menú lateral de la app, al cual podrá acceder en todo momento a partir de ahora (dado que la app es jerárquica plana). Tanto en el inicio como en el menú se muestran las opciones que puede realizar un usuario normal (cliente), las cuales son: Servicios cercanos (Volver al mapa), Buscar servicio (Busqueda del servicio por categoría), Solicitar servicio (Crear una petición) y Socios (apartado para administrar o conseguir membresía)

4. Convertirse en socio



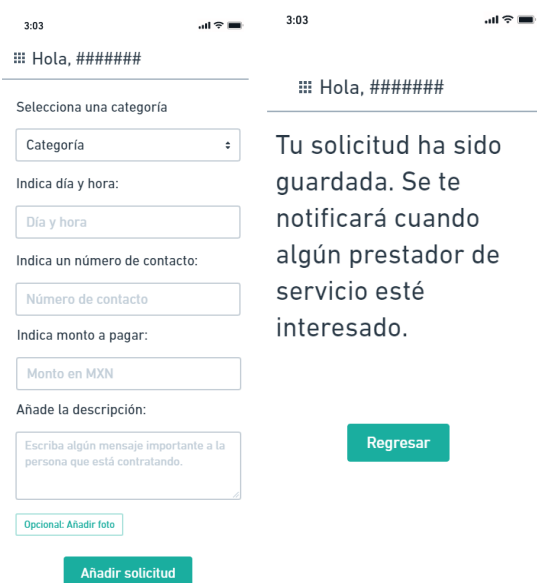
Esta opción puede presentarse cuando el usuario se esté registrando o bien, si el usuario no es socio y decide volverse socio en un momento posterior. En cualquiera de los casos, se le mostrará a los usuarios un apartado para subir una foto de su credencial de elector o tomarla con la cámara. Si se sube correctamente, se abrirá la pasarela de pago en la respectiva tienda con el fin de obtener la membresía, la cual podrá pagar mediante meses o años. Cabe destacar que no habrá otra forma de pago con el fin de no generar este servicio por completo. Una vez validado el pago, se regresará a la aplicación y se hará que el usuario lea los términos y condiciones de su contrato como socio. Una vez aceptado, se notifica que el pago fue aceptado y que está en proceso de revisión su identificación con el fin de validar la identidad de la persona. Cabe destacar que los pagos se administran directamente con la tienda, por lo que el control y la renovación será posible de seguirse en AppStore o Play Store.

5. Buscar servicios



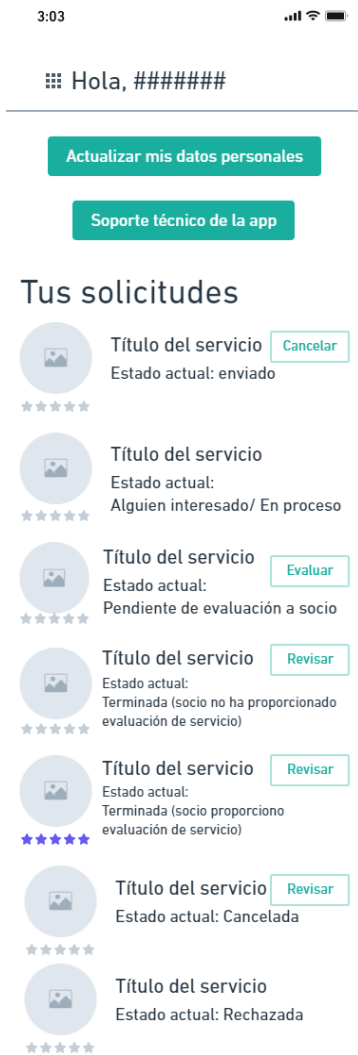
Al seleccionar la opción correspondiente en el menú o pantalla de inicio, se mostrarán las 12 categorías que tenemos para nuestra app. Al presionar sobre alguna de ellas, se mostrarán servicios relacionados a la categoría en cuestión acomodados de la más cercana a la más lejana y mostrando los detalles del servicio. Si alguna de las opciones desplegadas es de interés del usuario, se abrirá una vista ampliada del servicio a lo cual el cliente podrá solicitar. Al hacerlo, podrá indicar el día y hora que mejor le convenga dependiendo de las opciones desplegadas, indicar un número de contacto y mostrar dirección y referencias. Asimismo, puede añadir imágenes para hacer su caso aún más personalizado y que de esta forma, el socio le brinde un mejor servicio.

6. Solicitar servicios



En esta opción, el usuario puede crear alguna fuente de trabajo para los socios (es decir, solicitar un servicio). Para ello llena el formulario correspondiente con la categoría, día y hora que necesitaría el servicio, número telefónico, monto a pagar y descripción del servicio requerido, incluso colocando una imagen si lo cree conveniente. Tras registrarse estos datos en la base de datos, se le notifica al usuario que la petición fue almacenada, la cual podrá seguir mediante sus solicitudes en el panel correspondiente.

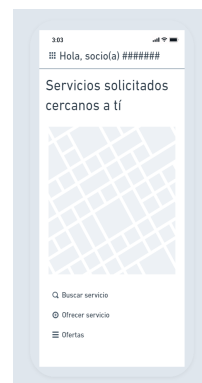
7. Administración del usuario y sus solicitudes



En este apartado, el usuario podrá darle seguimiento a las solicitudes de servicio y administración de su cuenta tras haber presionado su nombre y correo en el menú lateral de la aplicación. En primer lugar, se muestra un botón de actualización de datos personales que abrirá a una página web en la cual editará sus datos (solo celular, código postal, nombre y contraseña). Asimismo, también se muestra un botón que permite contactar al soporte técnico de la app para lo cual abrirá la aplicación de correo del dispositivo.

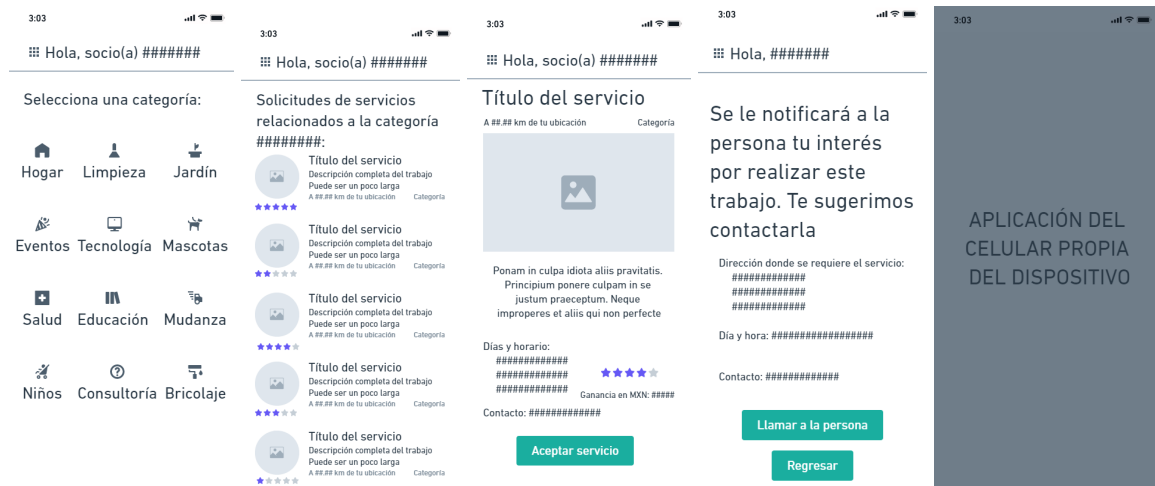
Asimismo, se muestra el listado de todas las solicitudes realizadas, mostrando uno de sus 7 posibles estados: Enviado, En proceso, Pendiente de evaluación a socio, Terminada sin evaluación de socio, Terminada con evaluación de socio, Cancelada y Rechazada. Asimismo, dependiendo del estado se mostrarán botones de cancelar (para detener la solicitud), evaluar (brindar calificación al socio) o revisar (estado general de la solicitud).

8. Panel de socios (exclusivamente socios)



Este apartado es similar al de los clientes, solo que en el mapa se despliegan las ofertas de trabajo para socios con el fin de conocer las necesidades en su zona. Se muestran las opciones que se tratan más adelante: Buscar servicios , Ofrecer servicios y Solicitudes.

9. Buscar servicios (exclusivamente socios)



Parecido a la del usuario normal. Al seleccionar la opción correspondiente en el menú o pantalla de inicio, se mostrarán las 12 categorías que tenemos para nuestra app. Al presionar sobre alguna de ellas, se mostrarán las peticiones de servicio relacionadas a la categoría en cuestión acomodadas de la más cercana a la más lejana y mostrando los detalles del servicio. Si alguna de las opciones desplegadas es de interés del socio, se abrirá una vista ampliada de la oferta de servicio a lo cual podrá aceptar trabajar. Al hacerlo, se le notificará a la persona el interés por realizar dicho trabajo y le aparecerá la dirección del cliente, así como sus datos de contacto. Se sugiere llamar por teléfono al solicitante con el fin de agilizar el proceso, por lo que se coloca un botón que lanza la aplicación de celular con el número de marcado.

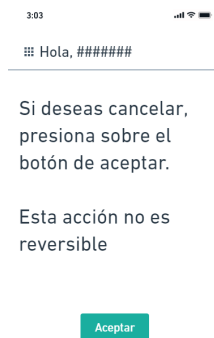
10. Ofrecer servicios (exclusivamente socios)

En esta opción, el usuario puede crear alguna fuente de trabajo para los socios (es decir, solicitar un servicio). Para ello llena el formulario correspondiente con la categoría, día y hora que necesitaría el servicio, número telefónico, monto a pagar y descripción del servicio requerido, incluso colocando una imagen si lo cree conveniente. Tras registrarse estos datos en la base de datos, se le notifica al usuario que la petición fue almacenada, la cual podrá seguir mediante sus solicitudes en el panel correspondiente.

11. Ofertas (exclusivamente socios)

Se muestra el listado de todas las solicitudes realizadas, mostrando uno de sus 7 posibles estados: Ofertada, En proceso, Pendiente de evaluación a cliente, Terminada sin evaluación de cliente, Terminada con evaluación de cliente, Cancelada y Rechazada. Asimismo, dependiendo del estado se mostrarán botones de cancelar (para detener la solicitud), evaluar (brindar calificación al socio) o revisar (estado general de la solicitud).

12. Cancelar



Al seleccionar una solicitud en específico, el socio o cliente podrá cancelar el proceso de la oferta. Esto solo es posible cuando está ofertada/enviada. La acción es irreversible, por lo cual se recomienda solo llegar a este caso en caso de ser exclusivamente necesario.

13. Evaluar cliente/socio

Son vistas equivalentes, solo que en cada caso se evalúa a alguien diferente. Se realiza cuando el servicio ya fue brindado y el objetivo es otorgar una calificación al usuario para que las personas de esta plataforma conozcan si es confiable o no trabajar/emplear a la persona.



3:03

Hola, socio(a) #####

Evaluación al cliente

Indica la evaluación para el cliente:

★★★★☆

Comentarios al cliente (opcional):

Comentarios al cliente

Enviar

3:03

Hola, socio(a) #####

Evaluación al socio

Indica la evaluación para el cliente:

★★★★☆

Comentarios al cliente (opcional):

Comentarios al cliente

Enviar

14. Revisar

En este apartado se muestra un resumen del servicio prestado/recibido. Se muestra la categoría, día y hora, descripción, ganancia, evaluación de ambas partes y comentarios del servicio.

15. Validar (Caso exclusivo para socios)

3:03

Hola, socio(a) #####

Título del servicio

A ##.## km de tu ubicación Categoría

Ponam in culpa idiotia aliiis pravitatis.
Principium ponere culpam in se
justum praeceptum. Neque
improperes et aliiis qui non perfecte

Días y horario:

#####

#####

#####

Ganancia en MXN: #####

Contacto: #####

Aceptar trabajo

Rechazar trabajo

3:03

Hola, #####

Revisión del servicio

Categoría

Día y hora:

00/00/00 00:00:00

Añade la descripción:

Mensaje con la descripción del trabajo realizado

Ganancia:

\$ 000000.00 MXN

Evaluación brindada

★★★★☆

Evaluación recibida:

★★★★☆

Comentarios del cliente:

Comentarios del cliente

Comentarios del socio:

Comentarios del socio

Estado actual: #####

Regresar

Mediante esta opción el socio podrá aceptar o rechazar solicitudes que se le hacen de manera directa. Al seleccionar la solicitud en cuestión, se muestra un resumen con los detalles propios del servicio del cliente solicitante, además de poder aceptar o rechazar

la propuesta de trabajo. En caso de aceptarla, se procede a contactar al usuario con el fin de establecer comunicación vía teléfono (tal como se hizo en el apartado de búsqueda de servicios para socios.

- **Wireframe en línea**

<https://whimsical.com/cornerworker-2k7qK4FHKCsM7ZFoB9wD9a>

- **Conclusiones del equipo**

Por Karla Najera:

A partir de este examen y el anterior me fue posible conocer la forma en la que se debe planear una sola app, la cual involucra demasiada planeación con el fin de obtener un producto de calidad y garantizar que sea lo más rentable posible.

Aunque como tal jamás programamos una app, considero que, de desearlo, la información aquí desglosada es una gran guía que nos permitirá darle una estructura a nuestra propuesta. Obvio tomando en cuenta que muchos de los aspectos aquí propuestos pueden variar con respecto a este documento debido a nuestra inexperiencia en el desarrollo de aplicaciones, pero no por ello significa que no se ha planteado una guía para codificar con un adecuado levantamiento de requerimientos.

Quizás lo único que considero que hizo falta fue agregar funciones para el envío de notificaciones mediante servicios como Firebase con el fin de afinar aún más el funcionamiento de la app. No obstante, en lo que respecta al resto de servicios y elementos de programación, opino que es una propuesta bastante novedosa y que tiene buenas herramientas (y novedosas) para su desarrollo.

Por lo anterior, considero que como examen (e incluso proyecto) pudimos consolidar los temas vistos en clase y usarlos en una aplicación real, que bien podría volverse en un rumbo al cual podemos dirigir nuestra carrera.

Por Alberto Cruz:

La realización del presente examen me ha permitido idear una forma de aplicar herramientas y conocimientos obtenidos a lo largo del curso, referentes a diferentes áreas del desarrollo de software. Este examen y todas las actividades que involucró, me han ayudado a tener una visión más general de todo el trabajo que conlleva desarrollar una aplicación, considerando aspectos no sólo técnicos sino también de otras áreas del conocimiento tales como el de marketing, financieros, diseño, hasta el esfuerzo intelectual que implica proponer una solución que satisfaga una necesidad en un determinado sector de la población.

Este trabajo me ayudó a conocer herramientas por medio de la investigación, a reforzar mis conocimientos sobre diferentes roles que intervienen en el desarrollo de software, tales

como la arquitectura, la infraestructura, el desarrollo y a su vez, los diferentes tipos de desarrollo como lo es el frontend, backend y las tecnologías asociadas.

En suma, el trabajo presentado en este documento, proporciona una excelente guía para poder desarrollar la aplicación en dado caso de seguir adelante con su implementación, puesto que abarca aspectos importantes tanto técnicos y no técnicos.

Por Facundo Acuña:

Al realizar los últimos dos exámenes con la dinámica de crear una app, nos dio una idea de la cantidad de elementos que se requieren para hacer una app. Aunque la metodología scrum es un estándar para desarrollar y entregar software, cuyas dinámicas como la 5W+2H para obtener las raíces para solucionar un problema, no deja exentos que se omitan algunos requerimientos que se requieran a la hora de implementar el desarrollo del producto.

También, considero que al momento de implementar el desarrollo es importante una buena comunicación entre los diferentes recursos humanos y los roles que representan, ya que una inadecuada comunicación puede tener como resultado un producto cuyos requerimientos funcionales no sean los que se hayan establecido en un inicio.

Por último, este trabajo me ayudó a conocer diversas herramientas de desarrollo que no conocía previamente, las metodologías de trabajo y la relación de los temas vistos en clase para poder realizar este trabajo, así mismo de considerar ciertos factores que nos dan un veredicto si es buena idea invertir o no para implementar una app. También que al momento de crear un app esta debe tener la capacidad de poder generar su propio capital para que se tengan ganancias y pueda sostenerse por sí misma, ingresos que le servirán para poder darle mantenimiento y cubrir gastos de operación.

● Referencias

Sebastian. (2019, 06 junio). *11 herramientas de desarrollo de software que te harán más productivo*. Ubiquim Code Academy.

<https://ubiquim.com/es/blog/20-herramientas-de-desarrollo-de-software-que-te-haran-mas-productivo/>

Subodh Dharmwan . (2021, 12 marzo). *Aplicaciones híbridas frente a aplicaciones nativas: la lista de verificación para descubrir cuál es la adecuada para usted*. Cynoteck.

<https://cynoteck.com/es/blog-post/hybrid-apps-vs-native-apps-the-checklist/>

Yasmani Tápanes. (2021, 13 septiembre). *Las 10 mejores herramientas de prueba de software*. Saasradar.

<https://saasradar.net/herramientas-prueba-de-software/>

Redacción KeepCoding . (2022, 13 enero). *Las mejores herramientas para el desarrollo de apps [Top 5]* . KeepCoding.

<https://keepcoding.io/aprender/herramientas-desarrollo-de-apps/>

Ekaterina Novoseltseva . (2012, 22 abril). *Principales herramientas de modelado de datos*. Apiumhub.

<https://apiumhub.com/es/tech-blog-barcelona/principales-herramientas-modelado-datos/>

Anónimo. (s.f.). *¿Qué es Google Maps Platform?*. Google Developers.

<https://developers.google.com/maps/faq?hl=es#what-is>

Anónimo. (s.f.). *Descripción general del SDK de Maps para Android*. Google Developers.

<https://developers.google.com/maps/documentation/android-sdk/overview?hl=es-419>

Anónimo. (s.f.). Maps SDK for iOS. Google Developers.

<https://developers.google.com/maps/documentation/ios-sdk/overview?hl=es-419>

Anónimo. (s.f.). *Uso y facturación del SDK de Maps para Android*. Google Developers.

<https://developers.google.com/maps/documentation/android-sdk/usage-and-billing?hl=es-419>