UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

Aplicación Para Crear Un Contacto Entre Emprendedores e Inversores (VVENTURE)

Daniel Alejandro Cabrera Checa

Ingeniería en Sistemas

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Sistemas

Quito, 07 de mayo de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Aplicación Para Crear Un Contacto Entre Emprendedores e Inversores (VVENTURE)

Daniel Alejandro Cabrera Checa

Nombre del profesor, Título académico **Fausto Pasmay, M.Sc.**

Quito, 07 de mayo de 2020

DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Nombres y apellidos: Daniel Alejandro Cabrera Checa

Código: 00110320

Cédula de identidad: 1715244230

Lugar y fecha: Quito, mayo de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

El presente trabajo describe el desarrollo de una aplicación móvil para Android y iOS que permite establecer una conexión entre emprendedores e inversionistas para, en un futuro inmediato, poder lograr una sociedad entre ellos. Para el desarrollo de esta aplicación se utiliza Flutter, ya que es un framework que nos ayuda a desarrollar aplicaciones nativas de Android y iOS desde una única base de código, se utilizan los diferentes servicios de AWS para alojar el servidor, el almacenamiento y la base de datos, tratando de lograr una arquitectura modular y de no dejar toda la carga al servidor principal. Esta arquitectura también ayuda a que la aplicación pueda escalar sin ningún problema.

Palabras clave: Flutter, Dart, Android, iOS, PHP, AWS, única base de código, S3, RDS, EC2.

ABSTRACT

This work describes the development of a mobile application for Android and iOS that allows establishing a connection between entrepreneurs and investors in order to achieve a partnership between them in the immediate future. For the development of this application Flutter is used, since it is a framework that helps us develop native Android and iOS applications from a single code base, the different AWS services are used to host the server, the storage and the database, trying to achieve a modular architecture and not to leave all the load to the main server. This architecture also helps the application to scale without any problem.

Key words: Flutter, Dart, Android, iOS, PHP, AWS, single code base, S3, RDS, EC2.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Según el reporte global del GEM del 2019/2020, Ecuador tiene un porcentaje sobre el 35% del TEA, ubicándose entre uno de los países de Latino América y el Caribe con mayor número de emprendedores entre 18 y 64 años (Bosma, Ionescu-Somers, Kelley, Levie, & Tarnawa, 2020). Uno de los retos más grandes para estos emprendedores es el de alcanzar una estabilidad económica, muchos de estos emprendimientos son impulsados por dicha necesidad. Esto implica que muchos de los emprendedores deben invertir mucho dinero para poder mantenerse a flote. Según el GEM por lo menos la mitad de los emprendimientos en Ecuador pueden salir adelante gracias a un apoyo económico (Bosma, et al 2020). Éste lo obtienen muchas veces de amigos, de su familia o de préstamos bancarios. Uno de los métodos más exitosos es el "crowdfunding" mediante plataformas como Indiegogo, GoFoundMe, o Kickstarter que se basan en vender su producto en una etapa previa a la de producción, de esta forma consiguen el financiamiento. Por otro lado, están los emprendedores que han obtenido este soporte económico mediante inversión de capital semilla, que bordea los 41 millones de dólares en el país, más del 84% ha sido colocado ya en emprendimientos previamente seleccionados.

Tomando en cuenta estos datos se decidió crear a VVENTURE, una aplicación móvil que pretende ayudar a emprendedores a conectarse con inversionistas. Los primeros van a poder crear un perfil con información de su emprendimiento que puede ser revisado por los inversionistas y animarlos a iniciar un contacto directo con el emprendedor; éste a su vez podrá analizar la información de los inversionistas para solicitar una revisión de su perfil.

El desarrollo de esta aplicación consta de dos partes: la primera es para el cliente y está instalada en el dispositivo del usuario; la segunda es el servidor y se desarrolla en diferentes módulos: computación, base de datos, almacenamiento y mensajería. El módulo de computación contiene toda la lógica del negocio y ayuda a la comunicación con el cliente; la

base de datos contiene las tablas necesarias para almacenar la información de los usuarios, el almacenamiento ayuda a guardar los archivos necesarios de los usuarios y algunos específicos de la aplicación, y la mensajería nos ayuda a enviar correos electrónicos a nuestra disposición. Esta arquitectura, al no tener todo en un solo módulo, nos ayuda a que la aplicación escale sin ningún problema.

DESARROLLO DEL TEMA

Formulación del problema

El problema de lograr un financiamiento de capital semilla se debe a de los inversionistas son gerentes de compañías grandes del país. El acceso a estas personas de posiciones importantes resulta difícil y muchas veces la comunicación se da a través de terceros como la AEI.

La forma en la que la AEI conecta a los emprendedores con los inversionistas es la siguiente:

- Ingreso del interesado al grupo de emprendedores de la AEI.
- Revisión de la información del proyecto.
- Participación en rondas simuladas para ver si el proyecto les interesaría a los miembros inversionistas.
- Servicio de mentorías pagadas para mejorar la preparación del emprendedor.
- De calificar, participación en una sola ronda con los miembros de inversionistas de la AEI.

De esta forma solo una cantidad muy pequeña de emprendedores van a ser vistos por los inversionistas.

Es entonces un proceso muy largo e incierto para los emprendedores, es muy difícil que los inversionistas puedan ver y analizar todos los proyectos disponibles y muchas veces la selección previa puede estar parcializada.

Descripción de la solución

La aplicación consta de una plataforma en la que un emprendedor puede poner la información más importante de su emprendimiento, este perfil sirve como una presentación del

proyecto para poder llamar la atención de un inversionista y motivarlo a iniciar un contacto. Al inversionista le permite dar a conocer la información más relevante de su compañía y sus opciones de ayuda, ya sea en forma de una mentoría o con la creación de una sociedad.

Casos de uso

Para la aplicación se tienen dos tipos de usuario: emprendedores e inversionistas. A continuación, en la Figura 1 se muestran los casos de uso y las limitaciones que cada usuario tiene en la aplicación.

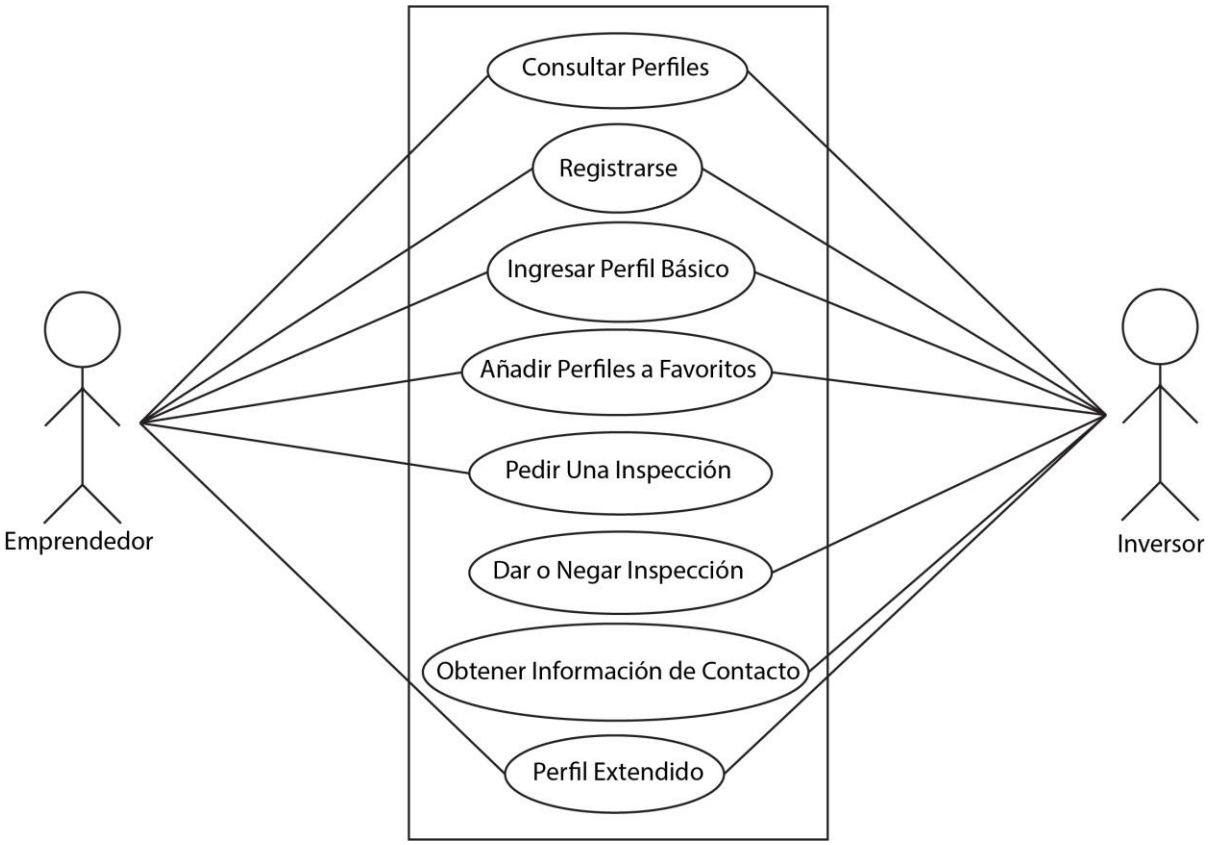


Figura 1 Casos de uso

Un emprendedor solo puede ver los perfiles de los inversionistas, y un inversionista solo puede ver el perfil de los emprendedores. Las funciones básicas de la aplicación como iniciar sesión y cerrar sesión se encuentran como parte de la funcionalidad de la aplicación. Por último, el perfil extendido del usuario cuenta con varias funcionalidades como: cambiar imagen de perfil, insertar o cambiar video de presentación, cambiar la información de perfil

básico, ingresar, editar y eliminar aspectos destacados, ingresar, editar y eliminar información extra detallada, ingresar y eliminar imágenes destacadas, e insertar y eliminar un cronograma del usuario.

Tecnologías utilizadas

En la siguiente sección se describen las tecnologías usadas para el desarrollo de la plataforma, dividiéndolas en las tecnologías utilizadas para el cliente y para el servidor.

Cliente. Para el cliente se utiliza un framework llamado Flutter el cual permite desarrollar aplicaciones nativas para Android y iOS desde una única base de código (flutter-dev, s.f). Flutter utiliza a Dart como lenguaje de programación, está enfocado en construir interfaces gráficas de una forma más sencilla e intuitiva. Una de las ventajas de utilizar Flutter es que permite ver los cambios de interfaz gráfica a tiempo real, ya que soporta tecnología JIT que inyecta los cambios a tiempo real al apk en el caso de Android y al runnable en el caso de iOS (flutter-dev, s.f). Para poder instalar un apk o un runnable de Flutter se debe tener un dispositivo móvil que soporte una arquitectura de x86_64, ARM de 86 y 64 bits. Flutter no soporta dispositivos que cuentan con arquitecturas x86 dejando de lado a más de un millón de dispositivos sin soporte. También el dispositivo debe tener Android Jelly Bean 4.1.x o superior, y para iOS 8 o superior. Para dispositivos de iOS tiene soporte desde iPhone 4s o superiores (flutter-dev, s.f).

Servidor. Para el servidor se utilizan las siguientes tecnologías.

Dot tk. Un dominio .tk gratuito para poder realizar llamadas y configuración del servidor de una forma adecuada.

EC2. Permite crear instancias de computación para poder montar servidores web de nuestra preferencia.

RDS. Ayuda a crear instancias de base de datos relacionales con diferentes tipos de DBMS, y permite configurarla con nuestras especificaciones.

S3. Es un servicio que ayuda a almacenar objetos de cualquier tipo, y sirve como CDN para el contenido de la aplicación.

AWS PHP API. Crea programáticamente Buckets de S3, consulta objetos a S3, ingresa y elimina objetos de S3.

Mailjet API. Maneja los servicios de mensajería de la aplicación. La utilización de Mailjet no solo ayuda a tener un diseño que pueda escalar de una forma más fácil, si no que ayuda a sobrepasar las limitaciones que AWS pone al puerto 25.

Arquitectura

Cliente. Para el cliente se utiliza el patrón de diseño de MVC. Donde el modelo es la representación de los objetos que van a ser mostrados, el view contiene a todos los widgets que van a ser renderizados por Flutter, y el controlador se encarga de la comunicación con el servidor, esta comunicación se la realiza mediante llamadas de HTTP y las respuestas del servidor serán siempre leídas en formato JSON. Toca recalcar que, para controlar el estado de cada uno de los widgets, es decir, para alertar cuando se realizó un cambio en el widget y para que Flutter lo vuelva a renderizar. Este se lo controla de manera individual en cada widget con el método de setState.

Servidor. Para el servidor se realizó un servicio REST para logar una comunicación estándar con el cliente. Este servicio REST solo va a responder llamados HTTP de POST y GET. No se implementó una arquitectura RESTFUL ya que PHP no soporta nativamente las llamadas de PUT, DELETE, o PATCH las cuales son parte de un servicio RESTFUL. Por el lado de la base de datos se utilizará PHP para poder comunicarse mediante el puerto 3306 a una base de datos remota de MySQL que está montada en una instancia de RDS. Para poder almacenar archivos como imágenes y videos se utiliza el API de AWS de PHP para que el almacenamiento no sea en el servidor si no que este se lo haga en dos Buckets de AWS S3 mediante llamadas de HTTP, y de la misma manera si se desea utilizar los servicios de email,

pero al API de Mailjet. En la Figura 2 se muestra cómo queda la arquitectura que maneja todos los servicios que VVENTURE utilizará.

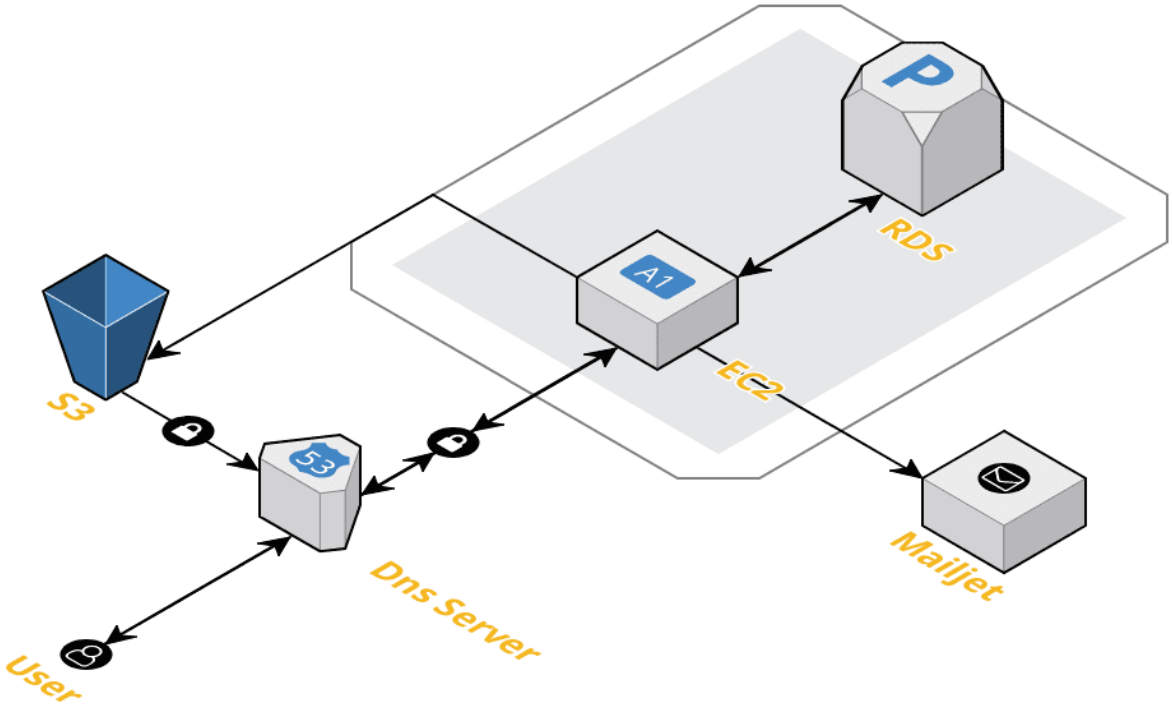


Figura 2 Arquitectura Servidor / Servicios

Interacción

La aplicación está dividida en tres módulos principales: inicio de sesión y registro, completar perfil básico, y aplicación principal. El módulo de aplicación principal se dividirá en varios submódulos para lograr la funcionalidad deseada.

Inicio de sesión y registro. En la Figura 3 se muestra un diagrama de flujo acerca del funcionamiento de este módulo.

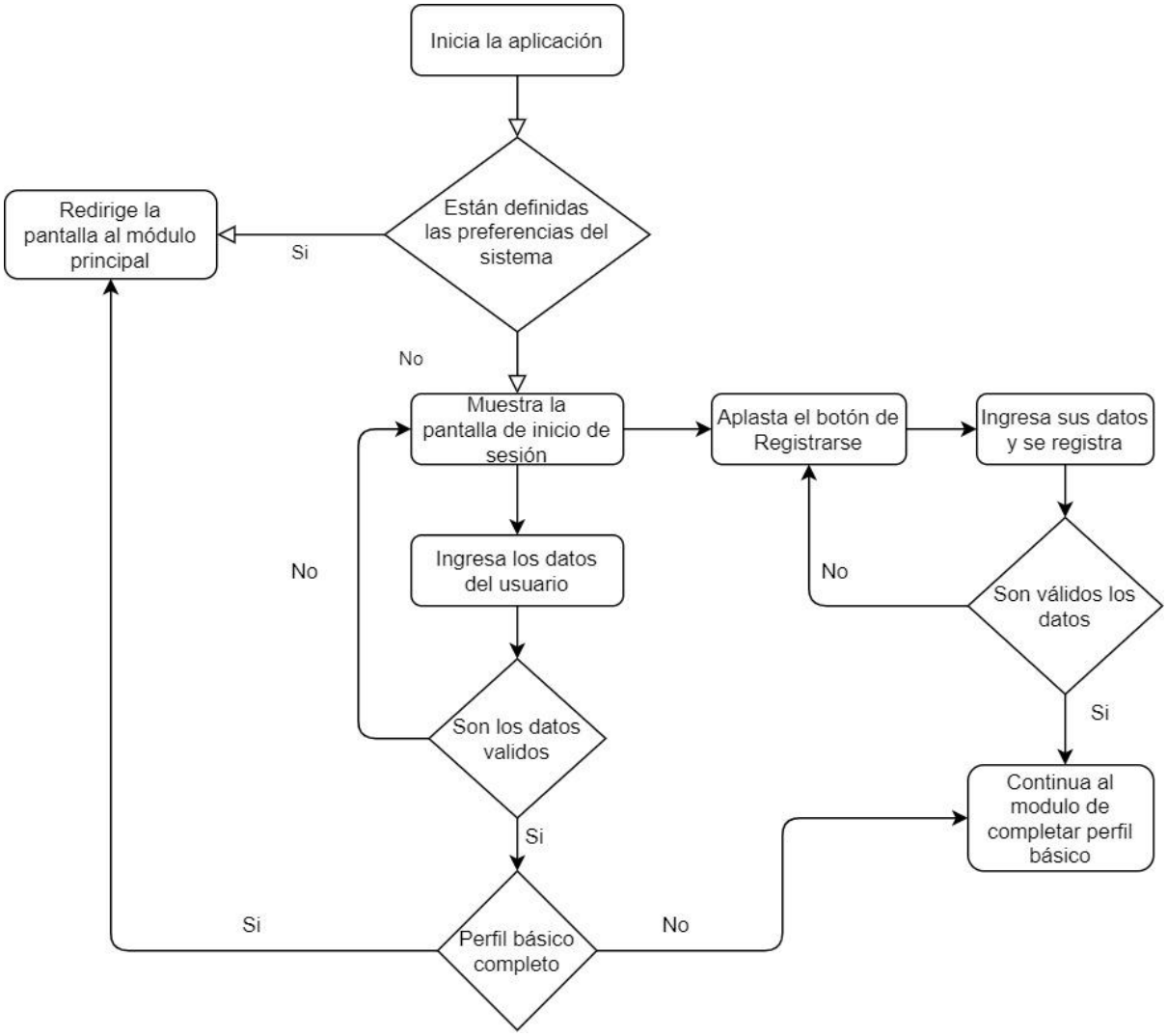


Figura 3 Diagrama de flujo inicio de sesión y registro

Este módulo se encarga de ver si el usuario ya inició sesión y de redirigirlo al módulo principal; si no lo ha hecho, este módulo deja al usuario iniciar sesión ya sea como emprendedor o como inversionista. También verifica que el usuario haya completado su perfil básico, ya que la aplicación no va a redirigir al usuario si no ha completado su perfil básico, la aplicación solo mostrará resultados de perfiles que tengan su perfil básico como mínimo. Este módulo también permite registrarse al usuario como emprendedor o inversionista. Al momento de ingresar los datos la aplicación mostrará si los datos están completos y si su usuario ha sido tomado. Una vez completado el registro la aplicación redirige al usuario al módulo de completar perfil básico.

Completar perfil básico. En la Figura 4 se muestra el diagrama de flujo acerca del funcionamiento de este módulo.

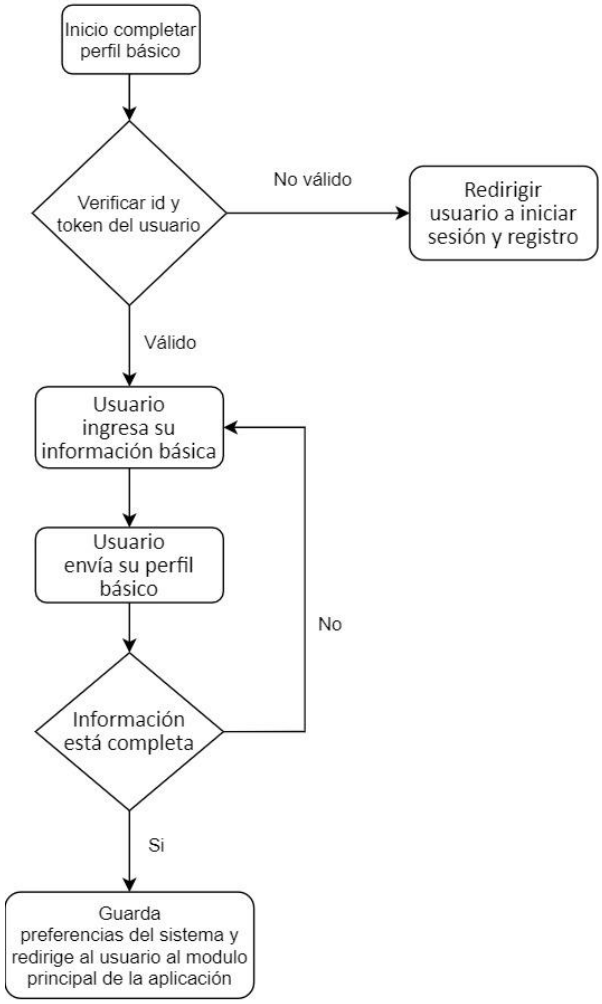


Figura 4 Diagrama de flujo completar perfil básico

En este módulo podremos completar nuestro perfil con información básica. La información que se tendrá que llenar será diferente dependiendo del tipo de cuenta que el usuario posea. Este módulo primero verificara que el usuario tenga un id, token y un tipo de cuenta válidos. Si esta información no es válida la aplicación redirigirá al usuario a la pantalla de inicio de sesión. Una vez que el usuario haya ingresado su información la aplicación mandara esta información al servidor para ser validada, si la información es válida se procederá a guardar el id, token y tipo de cuenta en las preferencias del usuario. Y si no lo es, la aplicación procederá a informar que la información no es válida y que se hagan los cambios respectivos.

Aplicación principal. En la Figura 5 se mostrará la funcionalidad básica de este módulo.

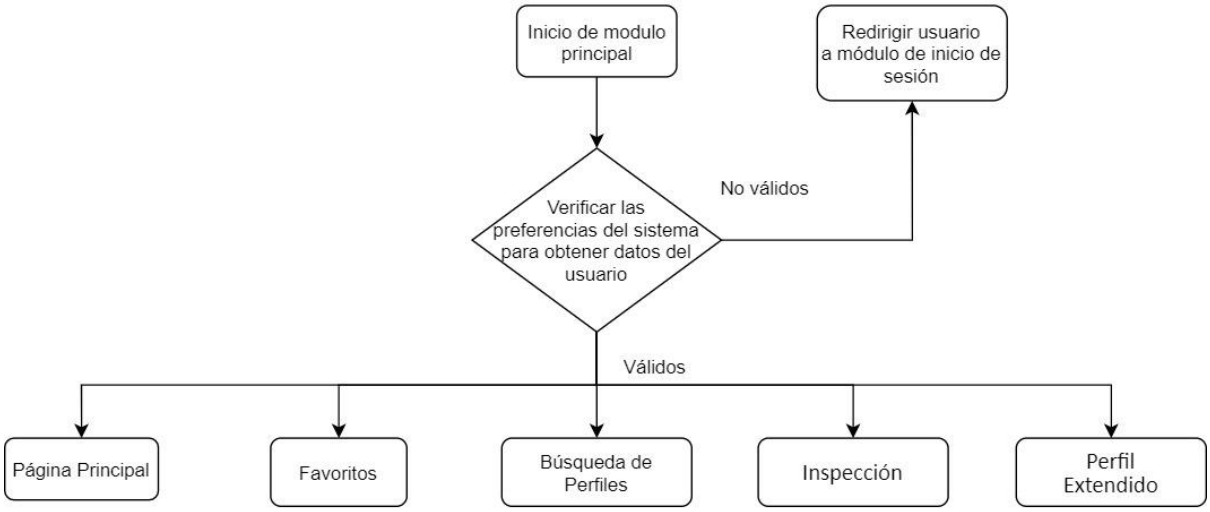


Figura 5 Diagrama de flujo funcionalidad básica modulo principal

Al inicializarse, este módulo chequea que los datos del usuario existan en las preferencias del sistema. Si no, el usuario es redirigido a la página de inicio de sesión; si son válidos los datos del usuario, este podrá navegar entre los diferentes submódulos. éstos son: página principal, favoritos, búsqueda de perfiles, inspección, perfil extendido.

Los submódulos de página principal, favoritos, búsqueda de Perfiles y perfil extendido, comparten el mismo diagrama de flujo tanto para el emprendedor como para el inversionista; mientras que el diagrama de flujo para inspección cambia dependiendo del tipo de cuenta. Los submódulos de página principal, favoritos, búsqueda de perfiles e inspección, en el caso del inversionista, comparten un módulo que nos permite ver en detalle los perfiles seleccionados; este módulo se denomina ver perfil.

Ver perfil. En este submódulo se ven los perfiles seleccionados, se pueden agregar a favoritos o quitar de favoritos al perfil que están visibles. Al final del perfil se encuentra un botón de acción, mismo que tendrá una interacción diferente, dependiendo del tipo de cuenta.

Si se está en la cuenta de emprendedor puede perderse al inversionista que no realice una inspección, mientras que si está en la cuenta de inversionista se puede pedir la información de contacto del emprendedor. En la Figura 6 se muestra el diagrama de flujo de este submódulo.

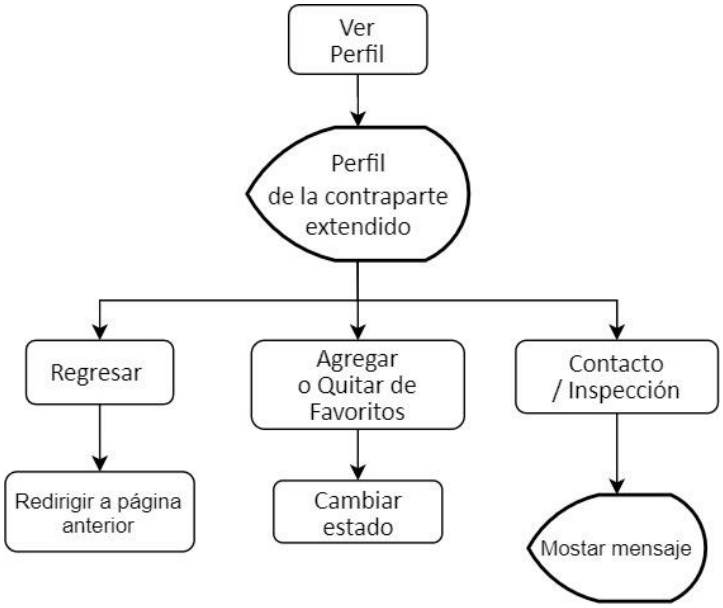


Figura 6 Diagrama de flujo ver perfil

Página principal. En este submódulo se seleccionan perfiles de la contraparte, es decir, un inversionista ve los perfiles de los emprendedores y viceversa. Se muestran todos los resultados de los perfiles y solo se muestra la información básica del usuario; solo se muestran los perfiles que hayan completado su registro al llenar su perfil básico. Una vez seleccionado un perfil, manda a la pantalla de ver perfil donde se ve el perfil del usuario seleccionado de forma extendida. En la Figura 7 podremos ver el diagrama de flujo relacionado con este submódulo.

Figura 7 Diagrama de flujo página principal

Favoritos. En este submódulo se ve una lista de usuarios marcados como favoritos en el submódulo de ver perfil. De la misma forma que en el submódulo de página principal, los usuarios solo muestran información básica del usuario, y su forma extendida en ver perfil. Se refresca la lista de usuarios arrastrando la pantalla hacia abajo. En la Figura 8 podremos ver el diagrama de flujo de este submódulo.

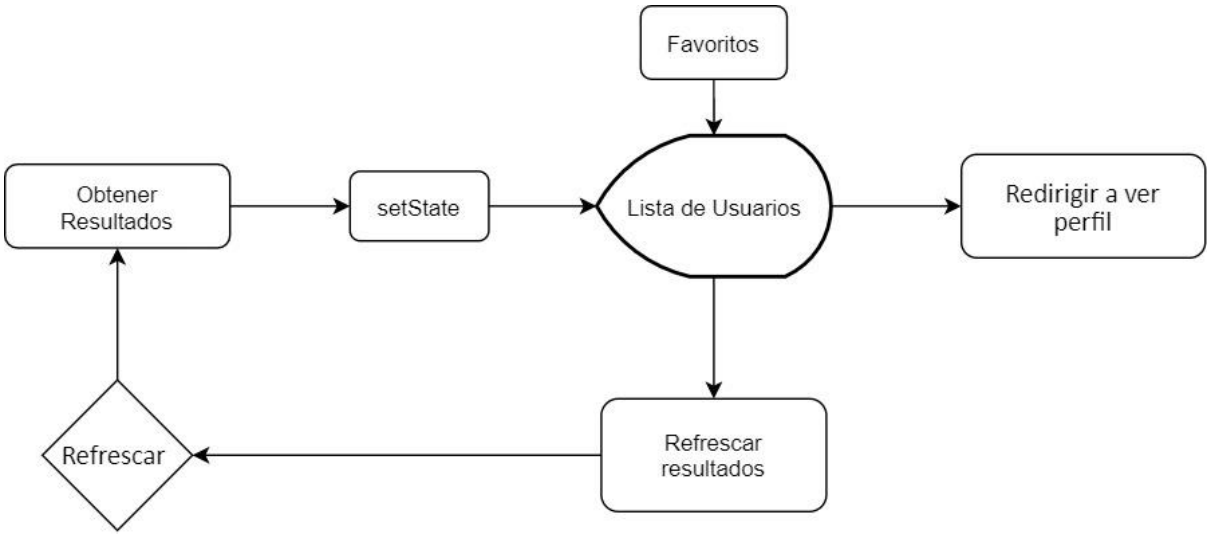


Figura 8 Diagrama de flujo favoritos

Búsqueda de perfiles. En este módulo los usuarios pueden buscar perfiles de sus contrapartes dado un criterio de búsqueda que ellos decidan poner. Se puede buscar por nombre de la organización, nombre y apellido del inversionista y emprendedor, intereses del inversionista, antecedentes del inversionista, etapa del emprendedor, problema del emprendedor, solución del emprendedor. En la Figura 9 se muestra un diagrama de flujo con la funcionalidad de este submódulo.

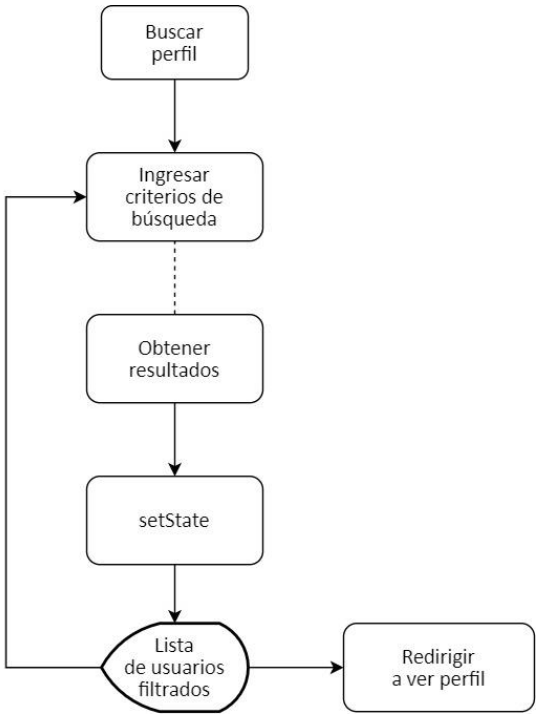


Figura 9 Diagrama de flujo búsqueda de perfil

Inspección emprendedor. En este módulo el emprendedor puede ver la retroalimentación de los inversionistas a los que se les ha perdido una inspección. A los inversionistas se les presenta en una lista y al seleccionar el inversionista se muestra una pantalla con información básica del usuario y la retroalimentación de este. El usuario podrá refrescar la información de este submódulo al arrastrar la pantalla hacia abajo. En la Figura 10 se mostrará el diagrama de flujo de este submódulo.

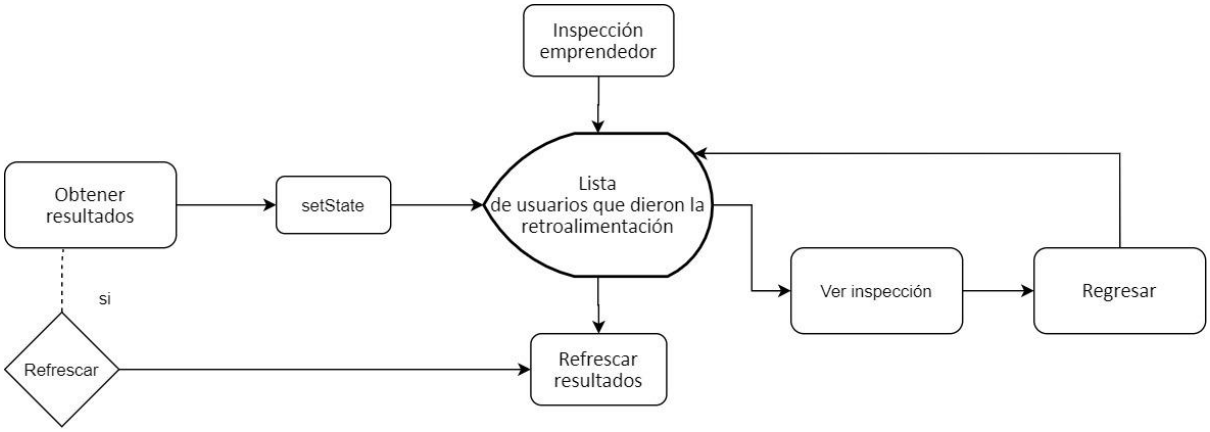


Figura 10 Diagrama de flujo inspección emprendedor

Inspección inversionista. En este módulo el inversionista puede ver una lista con los perfiles de los emprendedores que han solicitado una inspección. Al ver el perfil el inversionista puede decidir en dar o no una retroalimentación al emprendedor. El inversionista puede ver una lista de los emprendedores a los que ya ha dado una inspección y la retroalimentación que le ha dado al emprendedor. puede además refrescar la primera lista para ver nuevas inspecciones. En la Figura 11 se muestra el diagrama de flujo relacionado en este submódulo.

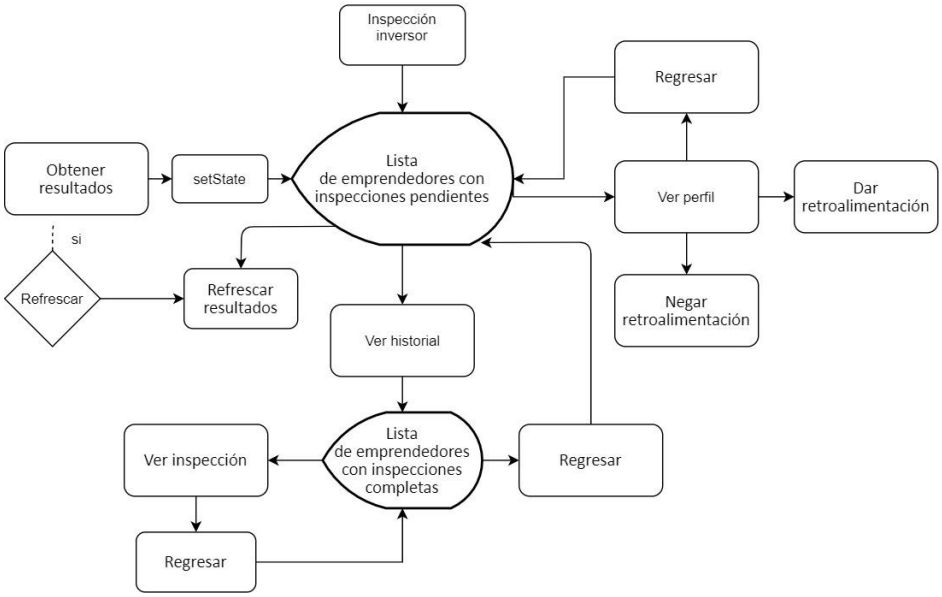


Figura 11 Diagrama de flujo inspección inversionista

Perfil extendido. En este submódulo el usuario puede ver y editar su perfil. El usuario pondrá toda la información básica de su perfil, ésta siendo: nombre de la organización, nombre y apellido del responsable, en caso del emprendedor (estado del proyecto, porcentaje que el emprendedor está dispuesto a dar, problema que está resolviendo, y su solución al problema), y en el caso del inversionista (sus antecedentes e intereses). El usuario podrá expandir su perfil al poder ingresar información relevante sobre su organización. La información que el usuario podrá expandir es la siguiente: momentos destacados, información extra, imágenes sobre su trabajo, línea de tiempo, y un video de presentación. El usuario podrá descargar su perfil en PDF y podrá cerrar sesión si este lo desea. En la Figura 12 se muestra el diagrama de flujo del funcionamiento de este submódulo.

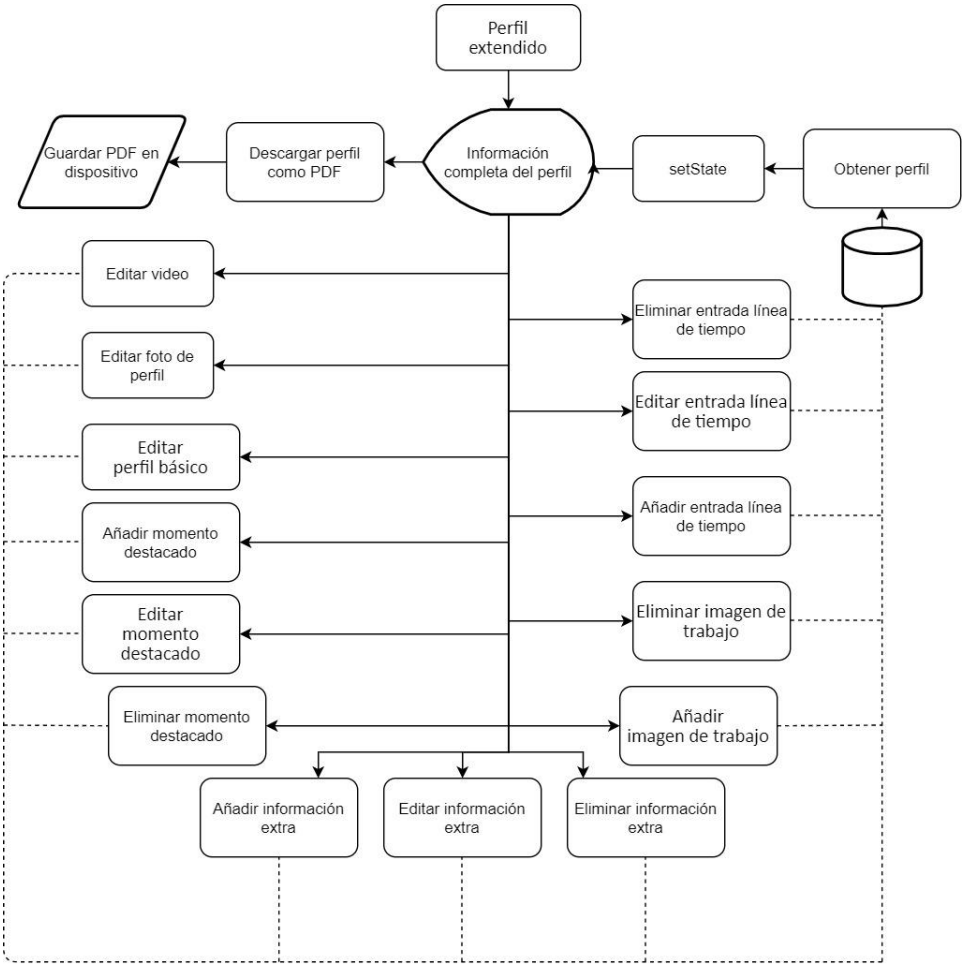


Figura 12 Diagrama de flujo perfil extendido

Trabajo futuro

Después de este ciclo de desarrollo se puede ver que la aplicación podría mejorar en algunos aspectos. Siendo estos los siguientes.

Incrementar el chequeo del sistema de comunicación cuando no existe internet, ya que por el momento hay que reiniciar el módulo que se está utilizando para poder iniciar la comunicación, debido a que la librería de Dart de HTTP no soporta nativamente una reconexión e intentar automáticamente esa comunicación con el servidor utilizando librerías de terceros como DIO.

Otro de los aspectos a mejorar seria la implementación de un diseño diferente de manejo de estado. Ya que actualmente como no se tienen muchos estados se pueden manejar desde los mismos widgets. Esto se puede solucionar implementando BLoC.

Además, se podría buscar una forma de poder cambiar los estados de los diálogos ya que estos, al no ser widget, para poder apreciar el cambio se los tiene que cerrar y volver abrir. por el momento no se puede cambiar el color de la barra de estado en iOS sin tener problemas que la barra no muestra los iconos del sistema.

Por último, se debería implementar un sistema de notificaciones, ya que al momento la aplicación no cuenta con uno.

CONCLUSIONES

El desarrollo de VVENTURE fue exitoso ya que se pudo obtener una aplicación nativa para Android y iOS. Gracias a la utilización de Flutter se pudo realizar un GUI moderno, muy importante para que el usuario pueda tener la misma experiencia en ambas plataformas. Flutter ayudó a realizar un desarrollo rápido para las dos plataformas, ya que al utilizar el mismo código base no se tuvo que desarrollar la misma aplicación en diferentes lenguajes de programación; y de esa manera agilitar el proceso de desarrollo.

Otra ventaja en el desarrollo de VVENTURE fue la implementación de un servicio REST. Ya que al tener un servicio REST ayudó a realizar la comunicación entre la aplicación y el servidor de una forma estandarizada mediante llamadas HTTP y las respuestas en formato JSON. También la utilización de Amazon Web Servicies ayudó a realizar una arquitectura escalable; si en un futuro pasaría a una fase de producción esta plataforma podría escalar sin ningún problema y podría brindar ese servicio a cientos de usuarios.

En el desarrollo de esta aplicación se encontraron varios inconvenientes y retos, siendo Flutter uno de éstos ya que, al utilizar Dart, se tuvo que aprender un nuevo lenguaje de programación, con la consiguiente pérdida de un poco tiempo al principio. Por otro lado, la edad de Flutter fue otro inconveniente pues, al no tener mucho tiempo en el mercado, su funcionabilidad es muy limitada. Para poder sobrepasar estas limitaciones es necesario el uso de librerías de terceros cuya documentación no es muy clara dándonos un retraso en el desarrollo de esta aplicación.

REFERENCIAS BIBLIOGRÁFICAS

Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., … Vrana, J. (s. f.). *PHP: PHP Manual - Manual*. Recuperado 20 de abril de 2020, de https://www.php.net/manual/en/

AWS. (s. f.-a). *AWS SDK for PHP 3.x*. Recuperado 18 de abril de 2020, de https://docs.aws.amazon.com/aws-sdk-php/v3/api/index.html

AWS. (s. f.-b). *Cloud Products*. Recuperado 16 de abril de 2020, de https://aws.amazon.com/products/

Bosma, N., Ionescu-Somers, A., Kelley, D., Levie, J., & Tarnawa, A. (2020, enero 1). *2019/2020 Global Report*. Recuperado 10 de abril de 2020, de https://www.gemconsortium.org/report/gem-2019-2020-global-report

Composer. (s. f.). *Namespaces | Composer API*. Recuperado 25 de abril de 2020, de https://getcomposer.org/apidoc/master/index.html

Dart. (s. f.). *Effective Dart: Documentation*. Recuperado 14 de abril de 2020, de https://dart.dev/guides/language/effective-dart/documentation

Flutter. (s. f.-a). *FAQ*. Recuperado 12 de abril de 2020, de https://flutter.dev/docs/resources/faq

Flutter. (s. f.-b). *Fetch data from the internet*. Recuperado 25 de abril de 2020, de https://flutter.dev/docs/cookbook/networking/fetch-data

Flutter. (s. f.-c). *Flutter Documentation*. Recuperado 16 de abril de 2020, de https://flutter.dev/docs

Let's Encrypt. (s. f.). *Documentation*. Recuperado 25 de abril de 2020, de https://letsencrypt.org/docs/

Mailjet. (s. f.). *Email API Overview*. Recuperado 20 de abril de 2020, de https://dev.mailjet.com/email/guides/

mPDF. (s. f.). *mPDF – mPDF Manual*. Recuperado 25 de abril de 2020, de https://mpdf.github.io/

MySQL. (s. f.). *MySQL :: MySQL 5.7 Reference Manual*. Recuperado 25 de abril de 2020, de https://dev.mysql.com/doc/refman/5.7/en/

Pub.dev. (s. f.-a). *chewie*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/chewie/latest/

Pub.dev. (s. f.-b). *Flutter Downloader*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/flutter_downloader/latest/

Pub.dev. (s. f.-c). *Flutter Timeline Widget*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/timeline_list/latest/

Pub.dev. (s. f.-d). *Image Picker plugin for Flutter*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/image_picker/latest/

Pub.dev. (s. f.-e). *path_provider*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/path_provider/latest/

Pub.dev. (s. f.-f). *permission_handler*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/permission_handler/latest/

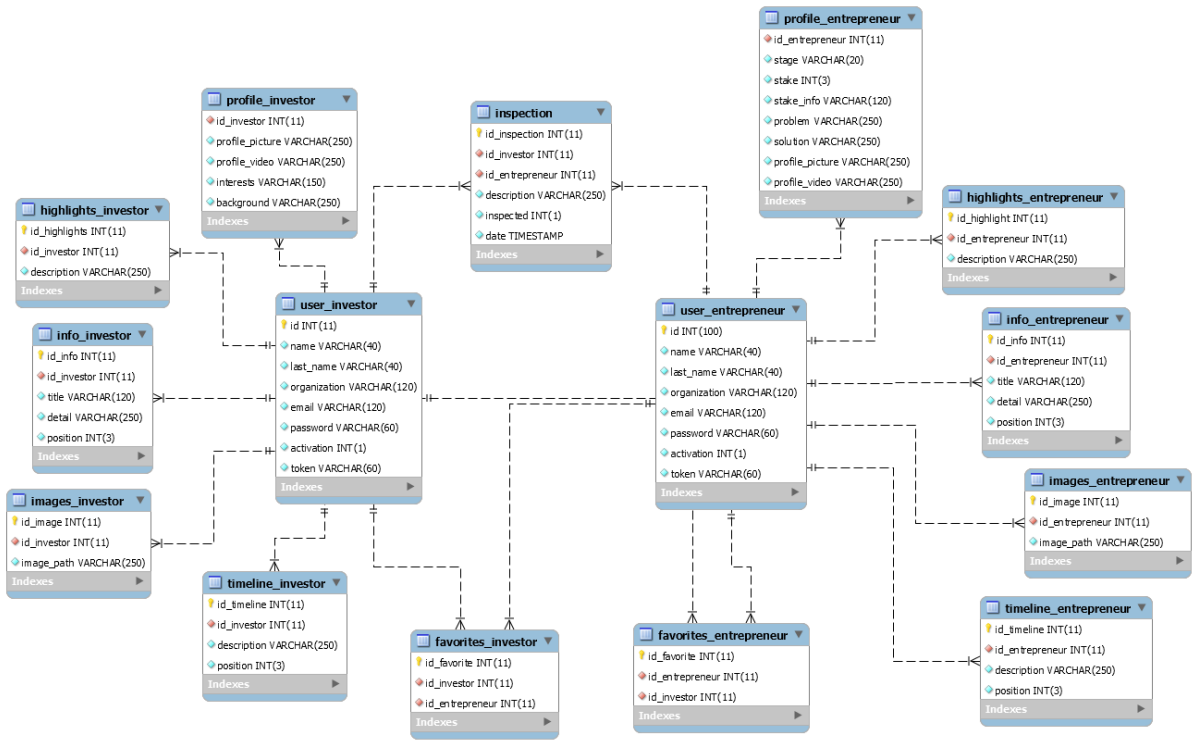
Pub.dev. (s. f.-g). *Shared preferences plugin*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/shared_preferences/latest/

Pub.dev. (s. f.-h). *snaplist*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/snaplist/latest/

Pub.dev. (s. f.-i). *Video Player plugin for Flutter*. Recuperado 25 de abril de 2020, de https://pub.dev/documentation/video_player/latest/

ANEXO A: ESTRUCTURA DE LA BASE DE DATOS

Para la base de datos se decidió utilizar una base de datos relacional. Y como DBMS se decidió utilizar MySQL. A continuación, se muestra el modelo entidad relación de la base de datos.



ANEXO B: DOCUMENTACIÓN DEL SERVICIO REST

Para poder obtener la información entre el cliente y el servidor se utilizará un servicio REST el cual fue escrito en PHP. Toda la información será enviada y recibida mediante llamadas HTTP con métodos POST y GET, toda la información será devuelta en formato JSON. A continuación, se mostrará la referencia para los endpoints disponibles.

Los términos comunes que van a ser utilizados a lo largo de la documentación son los siguientes:

- Auth (String): un token que sirve para verificar que el pedido fue enviado desde la aplicación.
- Type (int): tipo de cuenta del usuario. 1 para emprendedor y 2 para inversionista.
- Res (String): res es la respuesta del servidor. Si res contiene success esta es una llamada fue exitosa; caso contrario mostrara un error.
- Investor (int): investor es el id del inversionista.
- Entrepreneur (int): entrepreneur es el id del emprendedor.
- Inspection (int): inspection es el id de la inspección.

Endpoints

Iniciar sesión. En este endpoint podremos autenticar que el usuario se encuentre registrado en VVENTURE, y nos devolverá un token que identificará al usuario.

Estructura de url. https://vventure.tk/login/

***Método*.** Post.

***Argumentos*.** String ‘auth’, String ‘type’, String ‘email’, String ‘password’.

Respuesta. String ‘res’, String ‘type’, String ‘token’, ‘activation’.

Registro. Este endpoint nos ayudara a registrar a nuevos usuarios a VVENTURE. Después de registrar al usuario este nos devolverá un token para poder redirigir al usuario a que complete el registro.

Estructura de url. https://vventure.tk/register/

Método. Post.

Argumentos. String ‘auth’, String ‘type’, String ‘name’, String ‘last’, String ‘org’, String ‘email’, String ‘password’.

Donde name es el nombre del usuario, last es el apellido del usuario y org es la organización del usuario.

Respuesta. String ‘res’, String ‘type’, String ‘token’, ‘activation’.

Completar perfil básico emprendedor. Este endpoint nos permite completar el proceso de registro al completar el perfil básico del emprendedor.

Estructura de url. https://vventure.tk/complete_register/entrepreneur/

Método. Post.

Argumentos. String ‘auth’, String ‘token’, String ‘type’, base64 ‘image’, String ‘ext’, String ‘stage’, String ‘percentage’, String ‘exchange’, String ‘problem’, String ’solution’.

Donde image es la imagen en formato base64, ext es la extensión de la imagen, stage es el stado en el que se encuentra el emprendimiento, percentage es el porcentaje a ceder en la compañía, Exchange es porque se va a ceder ese porcentaje en la compañía, problem es el problema que busca resolver la compañía, solution es la solución de la compañía hacia el problema.

Respuesta String ‘res’, String ‘type’, String ‘token’, String ‘activation’.

Completar perfil básico inversionista. Este endpoint nos permite completar el proceso de registro al completar el perfil básico del inversionista.

Estructura de URL. https://vventure.tk/complete_register/investor/

Método. Post.

Argumentos. String ‘auth’, String ‘token’, String ‘type’, base64 ‘image’, String ‘ext’, String ‘interests, String ‘background’.

Donde image es la imagen en formato base64, ext es la extensión de la imagen, interests indica los intereses del usuario, background nos indica información relevante del inversionista.

Respuesta. String ‘res’, String ‘type’, String ‘token’, String ‘activation’

Resultados principales emprendedor. Este endpoint nos devuelve una lista con todos los inversionistas que están en la ampliación y han completado su perfil básico.

Estructura de url. https://vventure.tk/entrepreneur/results/

Método. GET.

Argumentos. String ‘auth’.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [name], [last], [organization], [image].

Resultados principales inversionista. Este endpoint nos devuelve una lista con todos los emprendedores que están en la ampliación y han completado su perfil básico.

Estructura de url. https://vventure.tk/investor/results/

Método. GET.

Argumentos. String ‘auth’.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [stage], [organization], [image].

Obtener información de contacto. Este endpoint nos permite pedir la información de contacto de emprendedor

Estructura de url. https://vventure.tk/investor/contact/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`, String `entrepreneur`

Respuesta. String ‘res’.

Anadir a favoritos emprendedor. Este endpoint nos permite agregar a un inversionista a la lista de favoritos.

Estructura de url. https://vventure.tk/entrepreneur/favorites/add/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `investor`

Respuesta. String ‘res’.

Anadir a favoritos inversionista. Este endpoint nos permite agregar a un inversionista a la lista de favoritos.

Estructura de url. https://vventure.tk/investor/favorites/add/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `entrepreneur`

Respuesta. String ‘res’.

Remover de favoritos emprendedor. Este endpoint nos permite remover un inversionista de la lista de favoritos.

Estructura de url. https://vventure.tk/entrepreneur/favorites/remove/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `investor`

Respuesta. String ‘res’.

Remover de favoritos inversionista. Este endpoint nos permite remover a un inversionista de la lista de favoritos.

Estructura de url. https://vventure.tk/investor/favorites/remove/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `entrepreneur`

Respuesta. String ‘res’.

Resultados de favoritos emprendedor. Este endpoint nos permite obtener una lista con los favoritos del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/favorites/remove/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [name], [last], [organization], [image].

Resultados de favoritos inversionista. Este endpoint nos permite obtener una lista con los favoritos del inversionista.

Estructura de url. https://vventure.tk/investor/favorites/remove/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [name], [last], [organization], [image].

Detalle de inspección emprendedor. Este endpoint nos permite obtener la retroalimentación del inversionista; y además nos devuelve información básica de este.

Estructura de url. https://vventure.tk/entrepreneur/inspection/detail/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token, String `inspection`, String `investor`.

Respuesta. String ‘res’, String `name`, String `last`, String `organization`, String `image`, String `detail`.

Lista de inspecciones emprendedor. En este endpoint podremos obtener una lista de los usuarios que han completado la retroalimentación al emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/inspection/inspections/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [name], [last], [organization], [image], [inspection].

Negar inspección inversionista. En este endpoint podremos negar una inspección del inversionista.

Estructura de url. https://vventure.tk/investor/inspection/deny/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `entrepreneur`, String `inspection`.

Respuesta. String ‘res’.

Detalle de inspección inversionista. Este endpoint nos permite obtener la retroalimentación que dio el inversionista; y además nos devuelve información básica de este.

Estructura de url. https://vventure.tk/investor/inspection/detail/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token, String `inspection`.

Respuesta. String ‘res’, String `name`, String `last`, String `organization`, String `image`, String `detail`.

Dar retroalimentación inversionista. En este endpoint podremos dar retroalimentación al emprendedor.

Estructura de url. https://vventure.tk/investor/inspection/feedback/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `entrepreneur`, String `inspection`, String `description`.

Respuesta. String ‘res’.

Lista de usuarios a los que ya se les ha realizado una inspección. Este endpoint nos permite obtener una lista con los usuarios que ya han sido inspeccionados.

Estructura de url. https://vventure.tk/investor/inspection/history/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [name], [last], [organization], [image], [inspection].

Lista de usuarios que han solicitado una inspección. Este endpoint nos permite obtener una lista con los usuarios que ya han solicitado una inspección.

Estructura de url. https://vventure.tk/investor/inspection/users/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [name], [last], [organization], [image], [inspection].

Buscar perfil emprendedor. Este endpoint nos devuelve una lista con los inversionistas que están en la ampliación y han completado su perfil básico, y que cumplen con el criterio que ha puesto el usuario.

Estructura de url. https://vventure.tk/entrepreneur/search/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`, String `query`.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [name], [last], [organization], [image].

Buscar perfil inversionista. Este endpoint nos devuelve una lista con los emprendedores que están en la ampliación y han completado su perfil básico, y que cumplen con el criterio que ha puesto el usuario.

Estructura de url. https://vventure.tk/investor/search/

Método. GET.

Argumentos. String ‘auth’.

Respuesta. String ‘res’, Array[users] ‘users`.

Users está conformado por los siguientes campos:[id], [stage], [organization], [image].

Eliminar momento destacado emprendedor. En este endpoint podremos eliminar un momento destacado del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/delete/highlight/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_highlight`.

Respuesta. String ‘res’.

Eliminar momento destacado inversionista. En este endpoint podremos eliminar un momento destacado del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/delete/highlight/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_highlight`.

Respuesta. String ‘res’.

Eliminar imagen de trabajo emprendedor. En este endpoint podremos eliminar una imagen de trabajo del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/delete/image/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_image`.

Respuesta. String ‘res’.

Eliminar imagen de trabajo inversionista. En este endpoint podremos eliminar imagen de trabajo del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/delete/image/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_image`.

Respuesta. String ‘res’.

Eliminar información emprendedor. En este endpoint podremos eliminar una entrada de información del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/delete/info/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_info`.

Respuesta. String ‘res’.

Eliminar información inversionista. En este endpoint podremos eliminar una entrada de información del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/delete/info/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_info`.

Respuesta. String ‘res’.

Eliminar entrada de línea de tiempo emprendedor. En este endpoint podremos eliminar una de línea de trabajo del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/delete/timeline/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_timeline`.

Respuesta. String ‘res’.

Eliminar entrada de línea de tiempo inversionista. En este endpoint podremos eliminar entrada de línea de tiempo del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/delete/timeline/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `id_timeline`.

Respuesta. String ‘res’.

Descargar perfil como pdf emprendedor. En este endpoint podremos generar un pdf y descargarlo inmediatamente.

Estructura de url. https://vventure.tk/entrepreneur/profile/download/profile/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`.

Respuesta. Archivo en pdf. Esta respuesta es la única que no es en formato JSON.

Descargar perfil como pdf inversionista. En este endpoint podremos generar un pdf y descargarlo inmediatamente.

Estructura de url. https://vventure.tk/inversionista/profile/download/profile/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`.

Respuesta. Archivo en pdf. Esta respuesta es la única que no es en formato JSON.

Obtener perfil extendido emprendedor. En este endpoint podremos obtener toda la información del perfil extendido del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/info/basic/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`.

Respuesta. String `res`, String `id`, String `organization`, String `name`, String `last`, String `image`, String `video`, String `stage`, String `stage`, String `stakeinfo`, String `problem`, String `solution`, Array `highlights`, Array `info`, Array `images`, Array `timeline`.

Obtener perfil extendido inversionista. En este endpoint podremos obtener toda la información del perfil extendido del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/info/basic/

Método. GET.

Argumentos. String ‘auth’, String `id`, String `token`.

Respuesta. String `res`, String `id`, String `organization`, String `name`, String `last`, String `image`, String `video`, String `interests`, String `background`, Array `highlights`, Array `info`, Array `images`, Array `timeline`.

Insertar momento destacado emprendedor. En este endpoint podremos agregar un momento destacado del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/insert/highlight/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail `.

Respuesta. String ‘res’.

Insertar momento destacado inversionista. En este endpoint podremos agregar un momento destacado del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/insert/highlight/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail`.

Respuesta. String ‘res’.

Insertar imagen de trabajo emprendedor. En este endpoint podremos insertar una imagen de trabajo del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/insert/image/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `image `, String `ext`.

La imagen debe ser en formato base64, y ext es la extensión de la imagen.

Respuesta. String ‘res’.

Insertar imagen de trabajo inversionista. En este endpoint podremos insertar una imagen de trabajo del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/insert/image/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `image `, String `ext`.

La imagen debe ser en formato base64, y ext es la extensión de la imagen.

Respuesta. String ‘res’.

Insertar información emprendedor. En este endpoint podremos insertar una entrada de información del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/insert/info/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `title `, String `detail`.

Respuesta. String ‘res’.

Insertar información inversionista. En este endpoint podremos insertar una entrada de información del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/insert/info/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `title `, String `detail`.

Respuesta. String ‘res’.

Insertar entrada de línea de tiempo emprendedor. En este endpoint podremos insertar una de línea de trabajo del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/insert/timeline/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail `.

Respuesta. String ‘res’.

Insertar entrada de línea de tiempo inversionista. En este endpoint podremos insertar entrada de línea de tiempo del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/insert/timeline/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail `.

Respuesta. String ‘res’.

Editar momento destacado emprendedor. En este endpoint podremos editar un momento destacado del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/highlight/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail`, String `id_highlight`.

Respuesta. String ‘res’.

Editar momento destacado inversionista. En este endpoint podremos editar un momento destacado del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/update/highlight/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail`, String `id_highlight`.

Respuesta. String ‘res’.

Editar información emprendedor. En este endpoint podremos editar una entrada de información del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/info/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `title `, String `detail`, String `id_info`.

Respuesta. String ‘res’.

Editar información inversionista. En este endpoint podremos editar una entrada de información del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/update/info/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `title `, String `detail`, String `id_info`.

Respuesta. String ‘res’.

Editar imagen de perfil emprendedor. En este endpoint podremos editar la imagen de perfil del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/profile_image/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `image `, String `ext`.

La imagen debe ser en formato base64, y ext es la extensión de la imagen.

Respuesta. String ‘res’.

Editar imagen de perfil inversionista. En este endpoint podremos editar la imagen de perfil del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/edit/profile_image/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `image `, String `ext`.

La imagen debe ser en formato base64, y ext es la extensión de la imagen.

Respuesta. String ‘res’.

Editar entrada de línea de tiempo emprendedor. En este endpoint podremos editar una entrada de línea de tiempo del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/timeline/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail`, String `id_timeline`.

Respuesta. String ‘res’.

Editar entrada de línea de tiempo inversionista. En este endpoint podremos editar una entrada de línea de tiempo del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/update/timeline/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `detail`, String `id_timeline`.

Respuesta. String ‘res’.

Editar video de perfil emprendedor. En este endpoint podremos editar el video de perfil del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/video/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `video `, String `ext`.

El video debe ser en formato base64, y ext es la extensión del video.

Respuesta. String ‘res’.

Editar video de perfil inversionista. En este endpoint podremos editar el video de perfil del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/edit/video /

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `video `, String `ext`.

El video debe ser en formato base64, y ext es la extensión del video.

Respuesta. String ‘res’.

Editar nombre emprendedor. En este endpoint podremos editar el nombre del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/name/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `name `, String `last`.

Respuesta. String ‘res’.

Editar nombre inversionista. En este endpoint podremos editar el nombre del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/update/name/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `name `, String `last`.

Respuesta. String ‘res’.

Editar organización emprendedor. En este endpoint podremos editar el nombre de la organización del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/organization/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `organization`.

Respuesta. String ‘res’.

Editar organización inversionista. En este endpoint podremos editar el nombre de la organización del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/update/organization/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `organization`.

Respuesta. String ‘res’.

Editar participación emprendedor. En este endpoint podremos editar la participación del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/stake/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `stake `, String `exchange`.

Respuesta. String ‘res’.

Editar etapa emprendedor. En este endpoint podremos editar la etapa de la organización del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/stage/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `stage`.

Respuesta. String ‘res’.

Editar problema emprendedor. En este endpoint podremos editar el problema de la organización del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/problem/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `problem`.

Respuesta. String ‘res’.

Editar solución emprendedor. En este endpoint podremos editar la solución de la organización del emprendedor.

Estructura de url. https://vventure.tk/entrepreneur/profile/update/solution/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `problem`.

Respuesta. String ‘res’.

Editar intereses inversionista. En este endpoint podremos editar los intereses de la organización del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/update/interests/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `interest`.

Respuesta. String ‘res’.

Editar antecedentes inversionista. En este endpoint podremos editar los antecedentes del inversionista.

Estructura de url. https://vventure.tk/inversionista/profile/update/background/

Método. POST.

Argumentos. String ‘auth’, String `id`, String `token`, String `type`, String `background`.

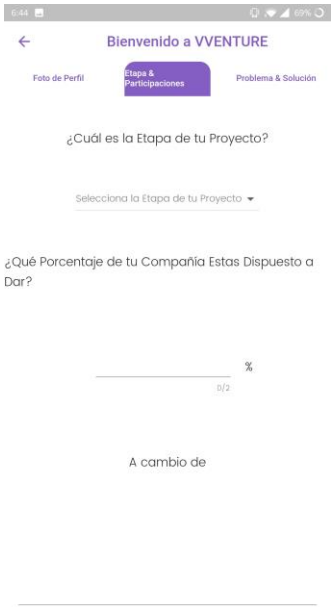
Respuesta. String ‘res’.

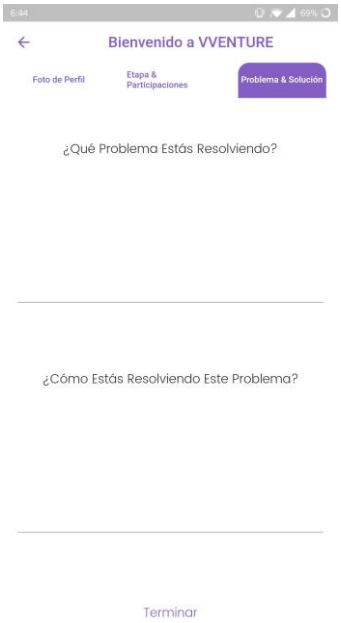
ANEXO C: GUI DE LA APLICACIÓN

Pantalla de iniciar sesión

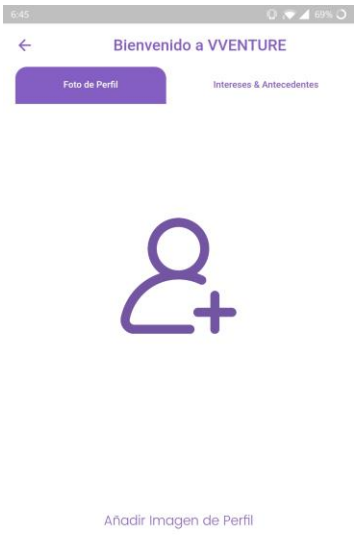
Pantalla de registro

Pantallas completar perfil básico emprendedor



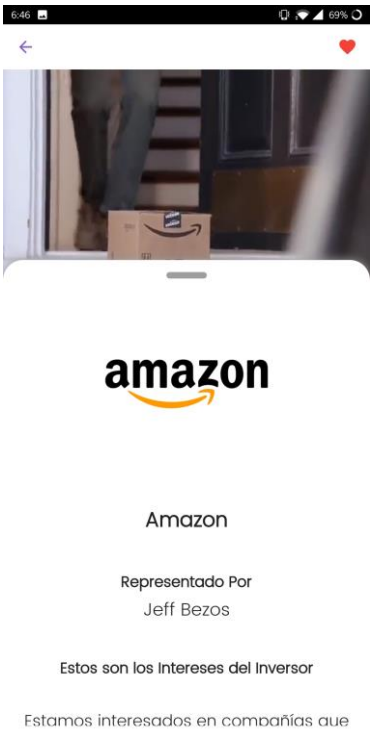


Pantallas completar perfil básico inversionista

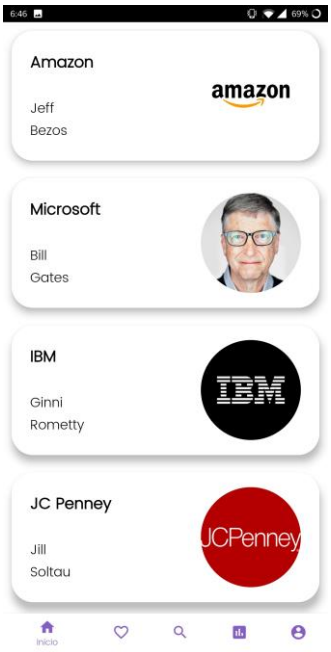


A continuación, tenemos las imágenes de la ampliación principal se pondrá solo las capturas del emprendedor ya que las del inversionista son muy similares. En caso de no ser similares se aclarará de quien es.

Pantallas de ver perfil

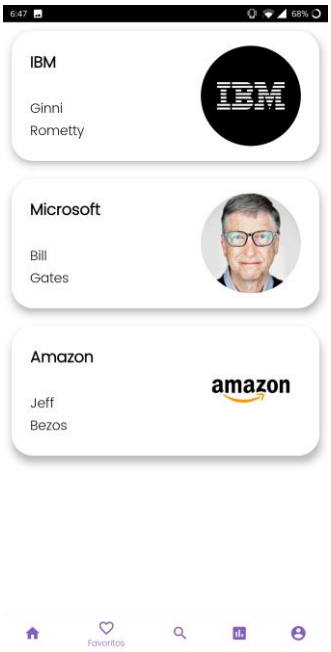


Pantalla de inicio

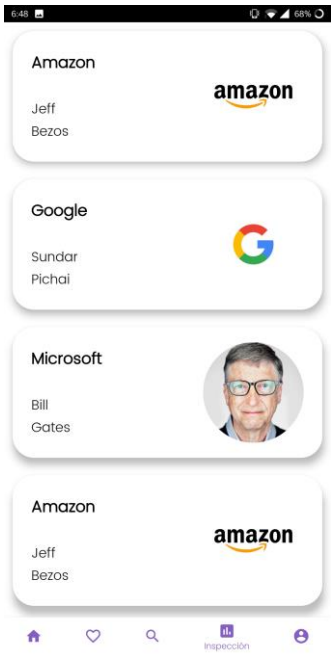


Pantalla de buscar perfil

Pantalla de favoritos



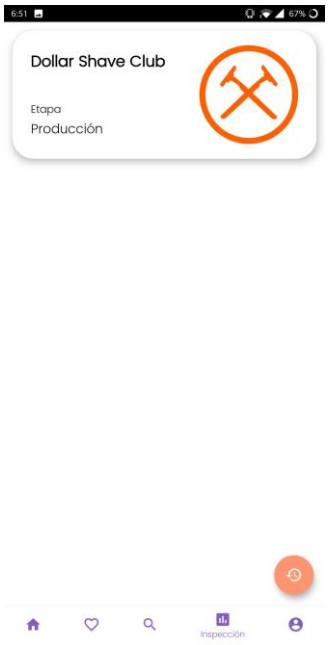
Pantalla inspección emprendedor



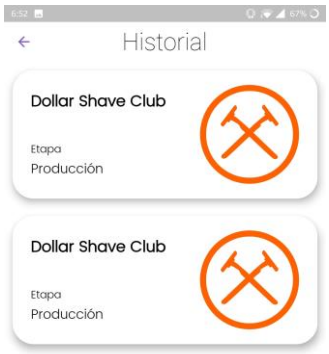
Pantalla detalle de inspección emprendedor

Pantalla dar revisión

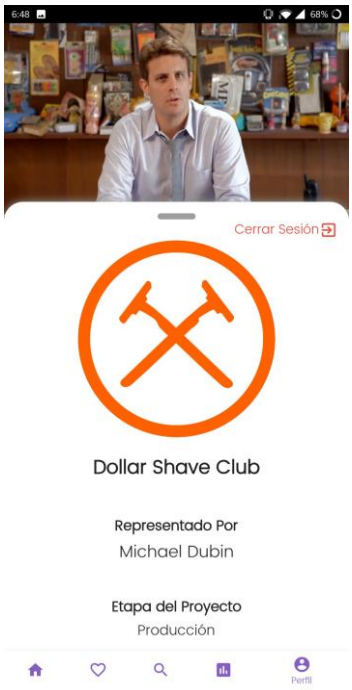
Pantalla inspección inversionista

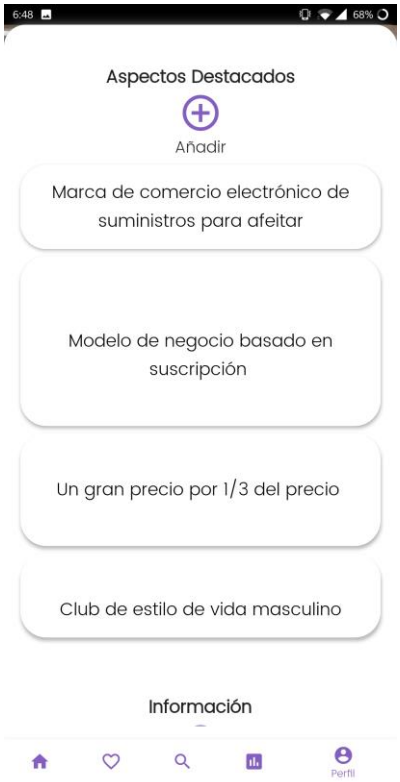


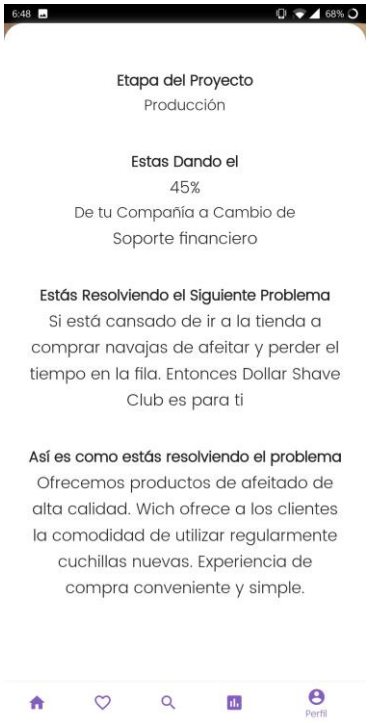
Pantalla historial de inspecciones inversionista



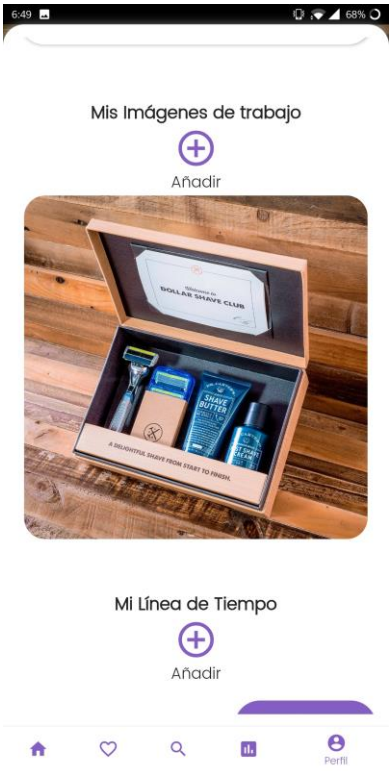
Pantalla perfil extendido con sus funcionalidades

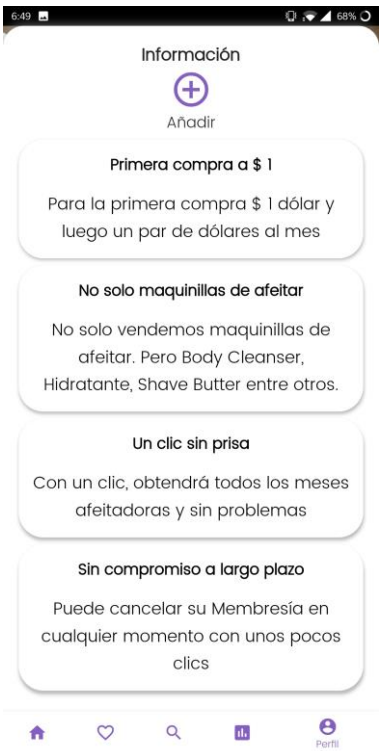







Aspectos Destacados
Añadir
ectrónico de
afeitar
Eliminar
Modelo de negocio basado en suscripción
Un gran precio por 1/3 del precio
Club de estilo de vida masculino
Información
Perfil


Mis Imágenes de trabajo
Añadir
Mi Línea de Tiempo
Añadir
Perfil


Información
Añadir
Primera compra a $ 1
Para la primera compra $ 1 dólar y luego un par de dólares al mes
No solo maquinillas de afeitar
No solo vendemos maquinillas de afeitar. Pero Body Cleanser, Hidratante, Shave Butter entre otros.
Un clic sin prisa
Con un clic, obtendrá todos los meses afeitadoras y sin problemas
Sin compromiso a largo plazo
Puede cancelar su Membresía en cualquier momento con unos pocos clics
Perfil


Mi Línea de Tiempo
Añadir
Fundado en 2011
En marzo de 2012, los inversores semilla aportaron $ 1 millón
De 2013 a 2015 lanzamos un montón de novedades para el mundo.
Para 2015, las ventas web de equipos de afeitado se
Perfil

Perfil descargado en pdf

ANEXO D: CÓDIGO FUENTE

El código fuente para VVENTURE se encuentra en https://github.com/danielcusfq o bajo el usuario danielcusfq. Este repositorio cuenta con el back end y el front end de VVENTURE.

Back end

El código para el back end se encuentra dentro del repositorio vventure_backend del usuario de github mencionado anteriormente. En esta carpeta tendremos subcarpetas que brindan la funcionalidad de la plataforma. También en esa misma carpeta se encuentra el archivo de connection.php el cual contiene la información correspondiente para la comunicación con la base de datos. Las otras carpetas son las siguientes: auth, complete_register, entrepreneur, investor, login, register, y validation.

Auth. Cuenta con un modelo para poder autenticar al usuario.

Complete_register. Completa el registro de ambos usuarios. También permite al usuario a completar su perfil básico.

Entrepreneur. Contiene la funcionalidad para el emprendedor de cada uno de los módulos mencionados en el desarrollo.

Investor. Contiene la funcionalidad para el inversionista de cada uno de los módulos mencionados en el desarrollo.

Login. Permite al usuario iniciar sesión y obtener un token de autenticación.

Register. Permite registrar al usuario ya sea como emprendedor o inversionista.

Validation. Permite al sistema verificar si un usuario es válido o no.

También es necesario que se tenga en el directorio principal los APIs de Mailjet, aws, y mpdf.

Front end

El código para el front end se encuentra en el repositorio de vventure_client del usuario de github mencionado anteriormente. En este repositorio contamos con la carpeta de vventure. En esta carpeta contamos con todo el proyecto de Flutter. Y en la carpeta de lib cuenta con todo el código de cada uno de los módulos mencionados en el desarrollo. La carpeta lib cuenta con los siguientes archivos principales y carpetas: main.dart, splash.dart, login, register, auth, basic_profile, entrepreneur, investor.

Main. Tiene todas las configuraciones principales de la aplicación como temas de la aplicación, aplicación principal y rutas.

Splash. En esta pantalla tenemos el mensaje de bienvenida y también verifica si el usuario ya inicio sesión o no y la redirige a su pantalla correspondiente.

Login. Cuenta con la funcionalidad de inicio de sesión.

Register. Permite al usuario a registrarse ya sea como emprendedor o inversionista.

Auth. Contiene un modelo que representa a la información de autenticación del usuario.

Basic_profile. Contiene la funcionalidad y las vistas necesarias para que el usuario pueda completar su perfil básico.

Entrepreneur. Cuenta con modelos, controladores, y views comunes para la funcionalidad de emprendedor, También en la carpeta de content se encuentra toda la funcionalidad de los módulos y submódulos mencionados en el desarrollo.

Investor. Cuenta con modelos, controladores, y views comunes para la funcionalidad de inversionista, También en la carpeta de content se encuentra toda la funcionalidad de los módulos y submódulos mencionados en el desarrollo.

También se necesitan los siguientes paquetes de dart/flutter instalados: chewie, cupertino_icons, flutter_downloader, http, image_picker, path_provider, permission_handler, shared_preferences, snaplist, timeline_list, video_player.

ANEXO E: GLOSARIO

AEI: Alianza para el emprendimiento e innovación del Ecuador.

Android: Sistema operativo para dispositivos móviles desarrollado por Google.

API: Interfaz de programación de aplicaciones.

APK: Un paquete para Android que contiene toda la información para distribuir una aplicación.

ARM: Un tipo de arquitectura de procesadores.

AWS: Amazon web services.

AWS PHP API: Api que nos permite integrar AWS con PHP.

BLoC: Bussines Logic Component es un patrón de diseño.

Buckets: Un sitio que nos permite almacenar archivos en AWS.

Crowdfunding: Es un mecanismo de financiamiento.

Dart: Lenguaje de programación orientado a objetos desarrollado por Google.

DBMS: Sistema de gestión de base de datos.

DELETE: Método de llamada HTTP.

DIO: Librería de Dart para llamados HTTP.

Dot tk: Proveedor de dominios gratuitos.

EC2: Amazon elastic computing cloud es uno de los servicios de AWS.

Flutter: Framework de desarrollo elaborado por Google.

Framework: Conjunto estandarizado de un entorno de trabajo.

GEM: Global Entreprenuership Monitor

GET: Método de llamada HTTP.

GUI: Interfaz gráfica de usuario.

HTTP: Protocolo de transferencia de hipertexto es un protocolo de transferencia de datos.

iOS: Sistema operativo desarrollado por Apple para sus dispositivos móviles.

Jelly Bean: version 4.3 de android.

JIT: Just In Time**.**

JSON: formato de texto para transferencia de información.

Mailjet API: Api que nos ayuda a enviar email desde Mailjet.

MVC: Model View Controller es un patrón de diseño de arquitectura de software.

MySQL: Sistema de gestión de datos relacional desarrollado por Oracle.

PATCH: Método de llamada HTTP

PDF: Formato para archivos digitales.

PHP: Hypertext Preprocessor es un lenguaje de programación de propósito general.

POST: Método de llamada HTTP.

PUT: Método de llamada HTTP

RDS: Amazon Relational Database Service es un servicio de AWS para poder crear instancias de bases de datos.

REST: Es una arquitectura de software para estandarizar la comunicación mediante llamadas HTTP.

RESTFUL: Es una arquitectura de software para estandarizar la comunicación mediante llamadas HTTP.

Runnable: Un paquete para iOS que contiene toda la información para distribuir una aplicación.

S3: Amazon Simple Storage Service es un servicio de AWS que nos permite almacenar objetos.

TEA: Total early-stage Entrepreneurial Activity.