DOCUMENTACIÓN COCTELPEDIA

Por Urko Urbieta Bueno, marzo-junio del 2020

Aclaración respecto al término 'Cóctel': Durante todo el desarrollo de la documentación, se refiere a cóctel como a cualquiera de las bebidas que puedan aparecer en la aplicación. Se les denomina así por el propio nombre de la app.

1. Identificación:

Coctelpedia es una aplicación para que diferentes usuarios, a la hora de salir de fiesta o escoger una bebida para tomar, no acudan siempre a las mismas opciones, y que así estas no sean repetitivas.

2. Antecedentes del proyecto (necesidades, utilidades, oferta actual):

Necesidades: A la hora de escoger un cóctel para tomar, los usuarios casi siempre recurren a los mismos, y eso hace que las fiestas, por este lado, se vuelvan monótonas, y esta aplicación busca crear una pequeña diferencia entre unas y otras, mostrando a los usuarios muchas opciones de bebidas a tomar, y a su vez sugiere algunas aleatoriamente. Por otro lado, en dichas fiestas se suele repetir también el hecho de querer jugar a ciertos juegos para estas ocasiones, y siempre se recurre a los que hay actualmente, que no tienen demasiada funcionalidad ni variedad.

Utilidades:

Oferta actual: Hay aplicaciones conocidas como «Piccolo» y «Game of shots», pero que simplemente tienen algún juego más elaborado, del cual si te aburres, tienes que descargarte otra aplicación. Hay otras aplicaciones que te dejan incluir tus propias frases, pero que nuevamente tienen poco contenido. Coctelpedia busca ser la «navaja suiza» de este tipo de aplicaciones, recopilando diferentes juegos, así como añadiendo la parte del listado de cócteles y la posibilidad de incluir tus propias frases al juego. En un principio no tiene más que unos pocos juegos, pero en futuras actualizaciones se añadirán muchos más.

3. Objetivos del proyecto:

Ofrecer una aplicación que cumpla lo que piden distintos usuarios, ya sea porque busquen el cómo preparar distintos cócteles, o porque quieran pasarlo en grande en la fiesta, ayudándose de juegos para todos los participantes. Como requisitos, en un principio estarían los de ofrecer una lista de cócteles que irá ampliando, y de forma similar, ofrecer una lista de juegos que irá también aumentando.

4. Secuenciación:

Se prevé que se necesitará un mínimo de diez semanas para desarrollar una versión estable de la aplicación, a la cual en un futuro se le añadirían actualizaciones que aumentarían su funcionalidad y la información que se da en la aplicación misma. Este espacio de tiempo se dividiría aproximadamente de la siguiente forma: en la primera semana, se llevaría a cabo la fase de análisis. En las tres semanas siguientes el diseño de la arquitectura y de la interfaz gráfica, y las seis semanas restantes irían más destinadas al 'back-end', así como el desarrollo de la base de datos, la api-rest y la aplicación de administración de las frases introducidas por los usuarios.

5. Recursos, herramientas y arquitectura.

El framework a utilizar es React Native, una solución sencilla para poder desarrollar la aplicación simultáneamente para Android y para iOS sin ni siquiera la necesidad de tener un iMac, aunque con la pega de que no estoy familiarizado ni con el lenguaje utilizado (Javascript) ni con el framework, que es totalmente nuevo para mí.

Las herramientas utilizadas serán: Visual Studio Code como editor, por su sencillez, y a su vez porque tiene numerosas extensiones que hacen más fácil el trabajo con el mismo; y Expo, como plataforma para desarrollar, probar y distribuir la aplicación, ya que hace que el desarrollo simultáneo para los dos sistemas operativos objetivos (iOS y Android) sea mucho más sencillo incluso, en comparación a usar React Native a secas.

El esquema del proyecto, sería, una aplicación móvil cliente con los mencionados juegos, datos sobre los cócteles y demás, mientras que del lado del servidor habría una API Rest que guardara las frases utilizadas en los distintos juegos en una base de datos, para que así los usuarios puedan enviar nuevas frases, y se actualice a los demás usuarios si dichas frases son aceptadas en el servidor. También se ha creado otra aplicación para administrar las frases que manden los jugadores para los juegos de Coctelpedia, y que así no vayan directas a la tabla del juego, y pase primero por una temporal, en la que el administrador valorara si esa frase debe entrar al juego o no.

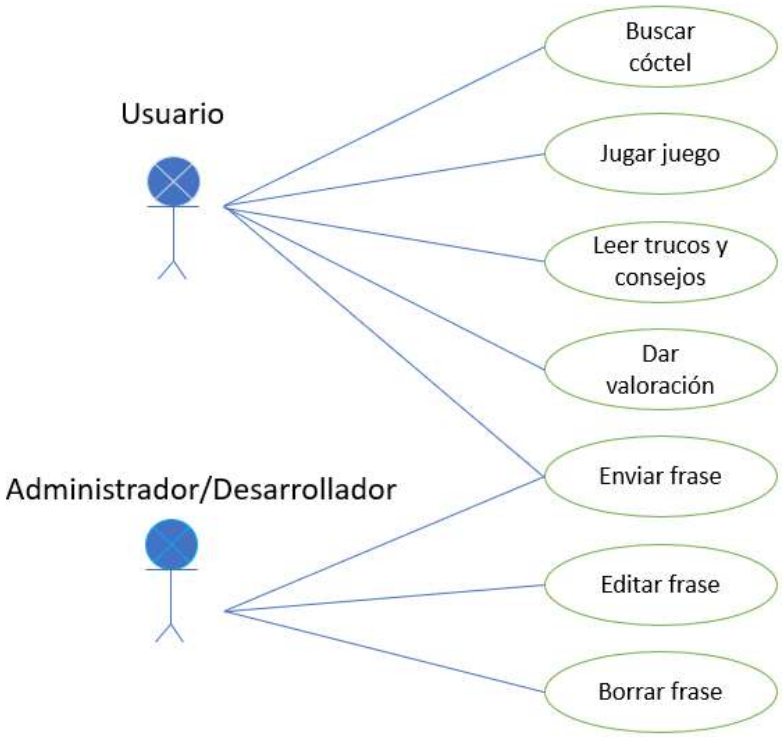
6. Fase de análisis:

El usuario estándar:

- Podrá buscar información sobre los distintos cócteles. Para encontrar uno que se asemeje a sus criterios puede realizar filtros y ordenar de distintas formas los resultados. A su vez podrá leer trucos y consejos acerca de la creación de mejores cócteles.
- Jugará a los juegos que desee entre los disponibles, con la cantidad de jugadores que él quiera.
- Si quiere, puede dejar una valoración de la app en la misma, o hacerlo en la tienda de aplicaciones cuando esté publicada. A su vez puede ponerse en contacto con los desarrolladores para enviar opiniones y sugerencias, así como reportar errores.
- Tendrá la opción de enviar frases para que se incluyan en el juego.

Los administradores/desarrolladores podrán:

- Enviar nuevas frases para ampliar los juegos.
- Editar las frases que hayan enviado los usuarios.
- Eliminar las frases no deseadas que hayan enviado los usuarios.



7. Fase de diseño:

Diseño de arquitectura:

Explicación sobre el diagrama: he valorado la posibilidad de hacer la estructura y de una forma sencilla enlazar las distintas partes del proyecto con flechas, pero teniendo en cuenta las múltiples conexiones que hay entre dichas partes, quedaba un tanto 'lioso', y he decidido poner las cuatro partes por separado y comentar en cada una de ellas con qué otras partes se comunicarán.

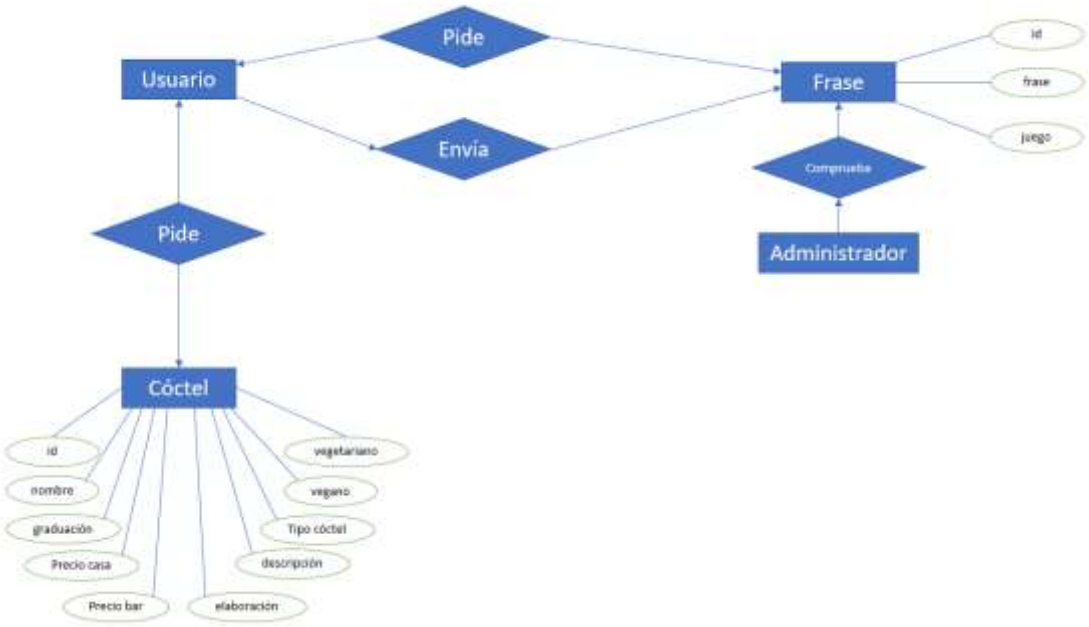
La aplicación del usuario final solamente se conecta a la API Rest, para enviar a una tabla temporal las frases que escriben los usuarios, y para enviar las valoraciones recibidas directamente a una tabla de valoraciones.

Coctelpedia Manager se conecta con la API Rest, para recibir los datos de la tabla de frases temporales, para poder editarlas, o eliminarlas, y las que se acepten se enviarán directamente a la tabla final de las frases del juego.

Base de datos que guarda las frases de los distintos juegos, y también las de las valoraciones recibidas por los usuarios.

Se comunica con las otras tres partes del proyecto: con la Base de datos para recibir y enviarle los datos de las aplicaciones, con Coctelpedia Manager para enviarle las frases temporales y recibir las definitivas, y con Coctelpedia para recibir las frases temporales y valoraciones, así como enviarle las frases definitivas para usarse en los juegos

Modelo entidad/relación:



En el modelo entidad relación vemos las posibilidades que tiene un usuario a la hora de usar la aplicación. A la hora de mirar la información de un cóctel el usuario recibe múltiples datos acerca de dichos cócteles. Por otro lado, vemos que el usuario puede pedir frases de los juegos, y también enviar nuevas.

Interfaz gráfica:

Pantalla principal con una breve presentación de la aplicación.

Listado de cócteles, con filtros en la parte superior, y parte de información en la zona inferior, en cada cóctel.

Información sobre un cóctel. Se muestran todos los datos, tales como descripción, elaboración y distintos precios.

Pantalla de juegos, en la cual se pueden también añadir jugadores y ver una bebida aleatoria (muestra la misma pantalla que la de información sobre cócteles, pero con un cóctel aleatorio)

Pantalla de juego: se muestra el reto en cuestión, la opción de ver la frase anterior y de sacar una nueva, así como el botón para mostrar una bebida aleatoria.

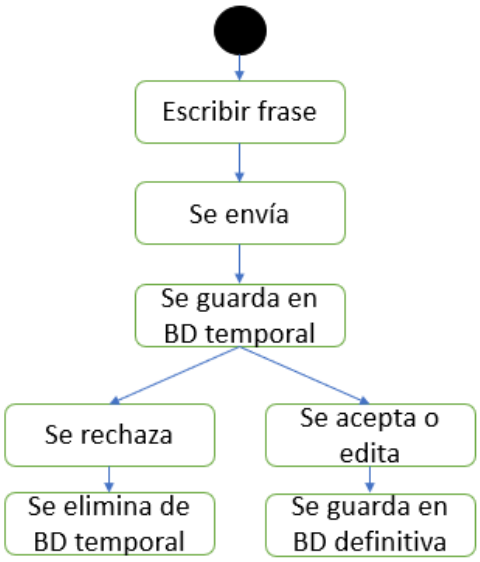
Pantalla para añadir, editar/eliminar jugadores (clickando sobre el nombre se accede a editar/eliminar)

Ajustes. Opciones para ponerse en contacto con el desarrollador, para compartir la aplicación, enviar una valoración y subir las frases que se le ocurran al jugador.

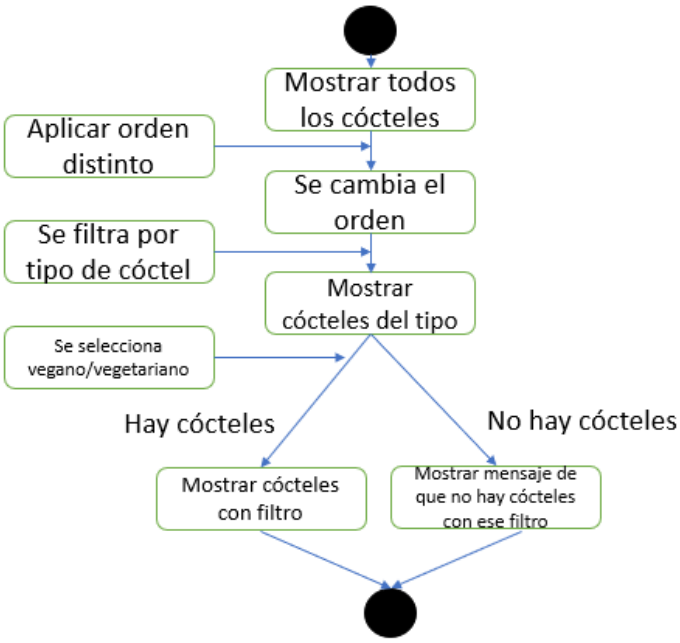
8. Fase de implementación:

Diagramas de actividad:

Enviar frase



Mostrar listado cócteles y filtrar



Memoria técnica:

En este apartado se nombran las funciones más importantes y lo que hacen.

En 'PlayersModal': getData, storeData, addPlayer, editPlayer, deletePlayer. Sirven para obtener los jugadores desde el almacenamiento del dispositivo, guardar los jugadores en el mismo, añadir nuevos, editar los actuales y eliminar el que se quiera respectivamente.

En 'StringHelper': getRandomString(prevStr, strList). Devuelve un string aleatorio de la lista strList, comprobando que no sea el anterior (prevStr).

En 'CoctelpediaFragment': changeStateVegan y changeStateVegetarian, cambian los valores de vegetariano y vegano entre true y false, para filtrar la lista. getInitialCoctel y getRandomCoctel devuelve un cóctel aleatorio para mostrar.

En 'GameChallengeFragment': getApiPhrases para obtener las frases desde la api, y si no coger las frases (sin actualizar) desde la memoria. getData para coger los jugadores de la memoria. getNext devuelve la siguiente frase o frase aleatoria. getPrev devuelve la anterior frase. getRandomCoctel devuelve un cóctel aleatorio.

En 'GamesFragment': checkPlayers comprueba que haya jugadores antes de entrar a un juego.

En 'SettingsFragment': emailPress muestra un popup para enviar un email al desarrollador. sharePress abre un modal para compartir la aplicación. starPress muestra las estrellas que se le han dado a la aplicación y envía la valoración a la base de datos. postPhraseToApi sube la frase ingresada por el usuario a la base de datos a una tabla temporal.

9. Fase de prueba:

(los registros de resultados, es decir, las soluciones para cada uno de los errores, se adjunta con el propio error)

Plan de prueba (errores encontrados):

- En los filtros del listado de cócteles, al iniciar la aplicación desde Android no aparecía el selector que daba las opciones para ordenar, ni tampoco el de filtrar por tipo de bebida. La razón era que en iOS y en Android no se puede usar el mismo tipo de selector o 'picker', y hubo que diseñar uno para Android.

- A la hora de recibir los jugadores desde el almacenamiento del dispositivo tuve varios problemas, ya que al colocarlos como 'estado', al reiniciar la pantalla de cualquier juego habiendo añadido nuevos jugadores, estos no aparecían. Para conseguir que se actualicen los jugadores, en vez de cargarlos en la pantalla misma, los he pasado como propiedad (algo similar a pasarlos como parámetro) y así no he tenido más problemas.

- Un nuevo problema que tuve con los estados en React Native fue para ordenar los cócteles en el listado. Es una forma un tanto extraña de trabajar para mí, el tema de los estados, y un fallo que les veo es que tardan cierto tiempo en actualizarse. El no saber que pasaba esto último provocó que pasara horas intentando conseguir que los estados funcionaran bien, hasta que hice que las propiedades se actualizaran en el *componentDidUpdate*.

- Al valorar la aplicación en ajustes, está preparado para abrir la tienda de aplicaciones. En Android lo hacía perfectamente, pero en iOS no abría la tienda ni el modal que se usa en este SO para hacer esta acción. El problema estaba en que aquí había que diferenciar el código para cada SO, y usando distintas funciones para abrir cada tienda de aplicaciones.

- Cuando cerraba los modales de 'editPlayer' y 'addPlayer', se generaba un pequeño parpadeo, el cuál se notaba un poco más en iOS. El problema estaba en que la capa de atrás de la interfaz, era un botón, el cual si se pulsaba cerraba el modal, y había que ponerle 'activeOpacity={1}' para que no parpadeara e hiciera su función.

- Un error muy tonto que tuve, fue que parecía que no podía guardar información en la memoria interna con 'AsyncStorage', y tras horas con ese problema, di con la solución: en el constructor estaba llamando a AsyncStorage.Clear(), lo cual, obviamente, borraba todo lo que hubiera en la memoria interna antes de que se mostrara dicha información.

Pruebas unitarias:

- En la aplicación he tenido que realizar pruebas unitarias, comprobando mediante logs en la consola si realmente estaba enviando datos o no a la Api, ya que con Postman si podía enviar los datos, mientras que en la app no. El problema estaba en que en Postman estaba usando el método 'x-www-form-urlencoded', y en la app tenía que convertir la información a este método.

- Intentando establecer una conexión con la Api Rest tuve que usar este tipo de prueba también para ver si el problema estaba en la Api o en la aplicación. Tras varias pruebas encontré el error, algo mucho más simple de lo que pudiera parecer: la IP de mi ordenador había cambiado, y simplemente cambiándola en las constantes de la aplicación todo funcionó.

- Desde CoctelpediaManager en un principio solo se subían las frases al juego 'Retos', y al hacer cambios para que se pudieran añadir a los dos juegos existentes actualmente, he tenido problemas al hacer el picker, ya que no guardaba bien el nombre de la tabla a la que había que subir la frase. Finalmente tras enviar logs de la tabla a la que estaba cambiando con el picker y ver que me estaba apareciendo como 'undefined', tuve que cambiar el valor que guardaba el picker en sí para que funcionara correctamente.