UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Título del Proyecto:

Listazo: Facilita tus compras

Autor:

Garcia Simarro, Isaac

Director:

Tomás Gironés, Jesús

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

Máster en Desarrollo de Aplicaciones sobre Dispositivos Móviles

Septiembre del 2015

Contenido

Introducción

Descripción del problema

Es comúnmente conocido que si realizamos listas de la compra antes de encaminarnos hacia el supermercado gozamos de múltiples ventajas. A quien no le molesta oír aquello de "¿Qué tenemos que comprar?" Pues tiene fácil solución si dedicamos unos minutos a la semana para planificar una lista de la compra y nos exigimos cumplirla. Obtendremos resultados fantásticos, nos quitaremos estrés y ahorraremos tiempo y dinero:

Ahorraremos tiempo porque:

- Con un solo viaje al supermercado (o supermercados) es suficiente para hacer las compras para toda la semana asegurándonos que no nos falta nada.

Ahorraremos dinero porque:

- Se hace la compra con una lista, basada en el menú semanal, lo cual permite evitar tentaciones y compras innecesarias.
- Si se eligen alimentos de temporada (básicamente frutas y verduras) la factura resultará más económica.

Y por supuesto se reduce el stress provocado por las preguntas del tipo ¿qué tenemos que comprar?, cuando estas en el supermercado ya que iras con la lista preparada.

Por ese motivo surge la creación de las aplicaciones móviles para realizar listas de la compra. Ya que es una forma muy fácil, asequible y cómoda de llevar la lista siempre contigo en todo momento, para poder añadir siempre que quieras diferentes artículos que te vengan en mente.

Hoy en día hay muchas aplicaciones en el Market relacionadas con las listas de la compra. Pero todas tienen el mismo problema funcionan de forma off-line y no se pueden agregar datos desde otros dispositivos ni compartirlos con otros usuarios, es decir, los mismos habitantes de una casa que realizan una compra semanal, un grupo de amigos que necesitan realizar una compra conjunta… no pueden tener una lista unificada donde todos puedan ir añadiendo productos.

Objetivos

Con este proyecto pretendo desarrollar una aplicación de compras compatible con dispositivos de smartphone y tablet, dirigiéndonos al sector de los supermercados. Se trata de un mercado que cada vez está más en auge ya que todo el mundo lleva un smartphone encima hoy en dia.

¿Por qué no poder realizar una lista de la compra en él? Cada día son más las aplicaciones que se crean para facilitarnos la vida en las labores rutinarias que todos debemos hacer y la aceptación de los usuarios para adaptarlas a su vida está siendo espectacular en los últimos años.

Ya existen varias aplicaciones dedicadas a la realización de listas de la compra como myShopi, ListOn, TuLista… La mayoría están focalizadas en la creación de listas de compra sin ofrecer ninguna función añadida. Además todas son en formato offline con el problema de que solo una persona puede añadir los artículos.

La forma de consolidar una ventaja competitiva de diferenciación para **Listazo** consiste en dirigirnos inicialmente a un segmento de público determinado, los usuarios que tienen una necesidad de realizar ciertas compras comunes, y aquí esta nuestra diferencia en la posibilidad de crear listas conjuntas que varios usuarios pueden consultar, añadir o modificar. Es decir, desde la app podrás crear una lista y compartirla con tantas personas como desees, siempre que hayan aceptado tu petición de amistad. Con esto conseguimos que cualquiera que se acuerde en un momento determinado, este donde este, de algo que hay que comprar, pueda actualizar la lista y todos sus miembros puedan verlo en tiempo real.

El objetivo básico de esta aplicación es facilitar al usuario la actividad de realizar la compra, evitando olvidos en el supermercado, con una acción tan sencilla como es llevar el móvil consigo mismo.

Además la característica que más nos diferencia de la competencia es que en nuestra aplicación cuentas con la posibilidad de compartir listas con otros usuarios de la aplicación, facilitándose así la coordinación entre estos usuarios que tienen una necesidad en una lista común.

También la aplicación te facilita el recordatorio de los productos que puedes necesitar ya que dispone de una lista de artículos ya creados que podemos seleccionar rápidamente con unos sencillos “taps” de pantalla.

Por otro lado en el caso de querer incluir un producto no añadido en dicha lista podemos crearlo como un nuevo artículo al que podremos añadirle su propia foto (que el propio usuario puedes seleccionar) y un nombre (el nombre del producto).

Por último, que no menos importante, contamos con la posibilidad de crear varias listas, lo cual puede ser muy útil en el caso de comprar en varios supermercados o en el caso de compartir ciertas listas con unos usuarios u otros.

Motivación

El motivo de la elección del presente tema para el Proyecto es muy variado. Por un lado me permite utilizar de forma amplia e imaginativa los conocimientos técnicos aprendidos a lo largo de este master. La mayoría de asignaturas aportan unos conocimientos técnicos “básicos”.

Por otro lado detecté la necesidad de este tipo de aplicación móvil personalmente, recientemente cuando comencé a encargarme de las compras de una casa y no había ninguna aplicación que cubriera mis necesidades o si la había, era de pago.

Por este motivo ha sido por el cual me decidí a crear esta aplicación móvil ya que veo una deficiencia en el mercado y además veo que es una aplicación en la que puedo utilizar varios de los conocimientos aprendidos en master como las notificaciones push, distintos diseños de layouts, las nuevas listas RecyclerView que son más eficientes que los anticuados listviews…

Tecnologías utilizadas

Para poder tener las listas sincronizadas y compartidas entre distintos usuarios, se ha tenido que realizar un backend muy extenso y con bastante lógica. La tecnología utlizada ha sido NodeJS ya que últimamente se esta hablando mucho sobre el y es una tecnología muy demandada en el entorno laboral, por lo que he visto una buena forma de aprender e iniciarme en el sector. Además, NodeJS es 10 veces más rápido que PHP ya que todas sus llamadas son asíncronas por lo que puede estar procesando muchas peticiones al mismo tiempo, incrementando mucho la velocidad.

NodeJS se compone de módulos por lo que es muy extensible ya que si necesitas hacer algo nuevo tan solo tienes que instalar el modulo correspondiente y hacer uso de el.

Por ejemplo, para las notificaciones push existe un módulo llamado “node-gcm” que se encarga de gestionar todo el proceso.

Un posible ejemplo de uso sería el siguiente:

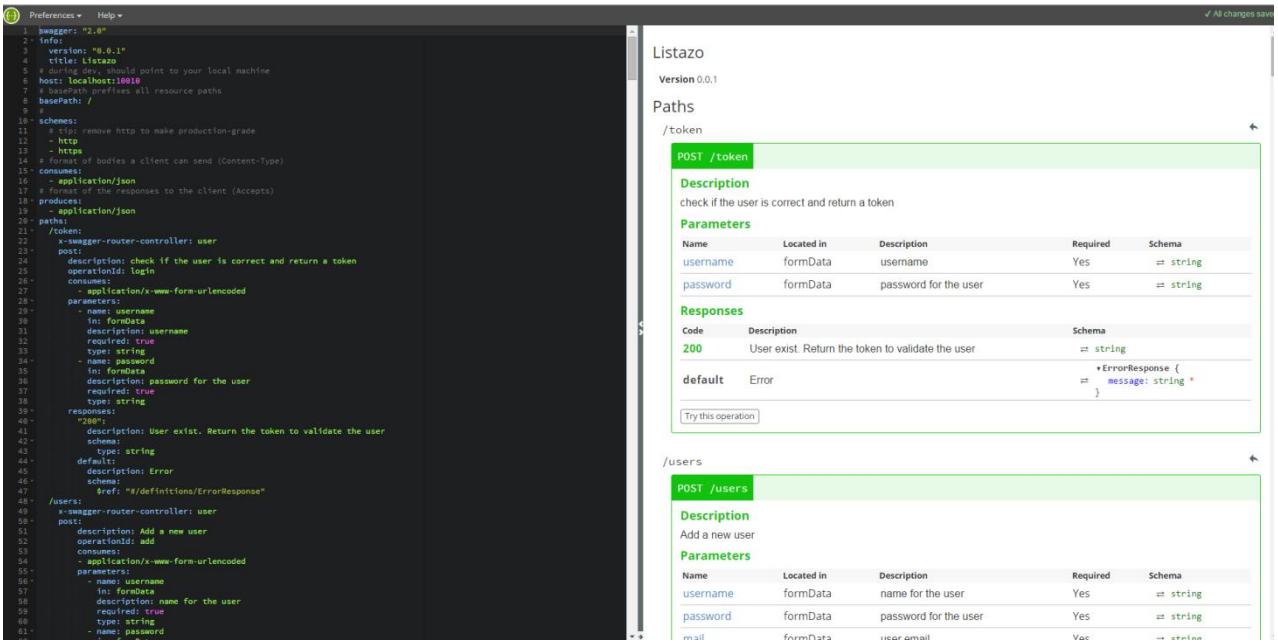
```
var gcm = require('node-gcm');

var message = new gcm.Message();

// Datos que se quieran enviar
message.addData('type', '1');
message.addData('idFriend', idUser);

// array de ids de los dispositivos a los que enviar la notificación
var regIds = ['REG_ID'];

// Api key del proyecto que implementa el GCM
var sender = new gcm.Sender('CONSTANTS.API_KEY');

// Ahora con sender enviamos
sender.send(message, regIds, function (err, result) {
    if(err) console.error(err);
    else    console.log(result);
});
```

Como se puede observar, el funcionamiento es muy sencillo. Creamos un mensaje, le añadimos los datos que queramos, indicamos a quien queremos enviarlo y por último se crea un objeto sender con nuestro API KEY y lo enviamos con su método send.

Todo el proyecto de node va incluido en el código de la aplicación para posibles revisiones.

También se ha utilizado swagger un potente framework para realizar apis que te crea automáticamente el checkeo de campos obligatorios en las peticiones así como la documentación.

Aquí se puede ver un fragmento de la especificación del API de Listazo:

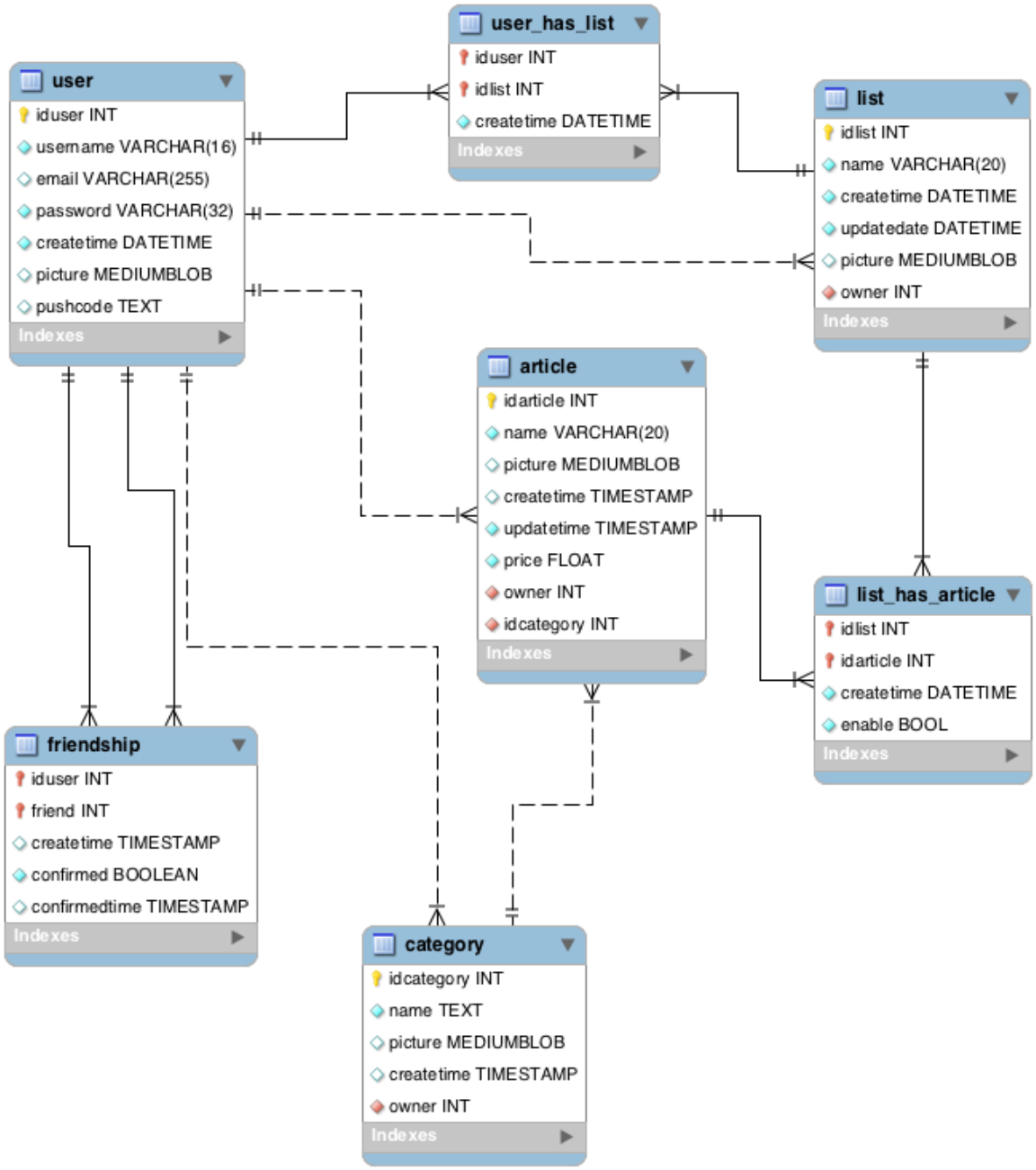


Por último comentar que tanto la base de datos como el servidor donde se está ejecutando el api en Node, están alojados en los servidores de amazon web services, ya que este te proporciona una capa básica de sus servicios durante un año de forma totalmente gratuita.

Arquitectura de la aplicación

Modelo de datos

Para la base de datos he utilizado MySQL ya que es un gestor rápido y sencillo que conozco bastante bien. Además, con otro módulo se puede usar facilmente desde node.



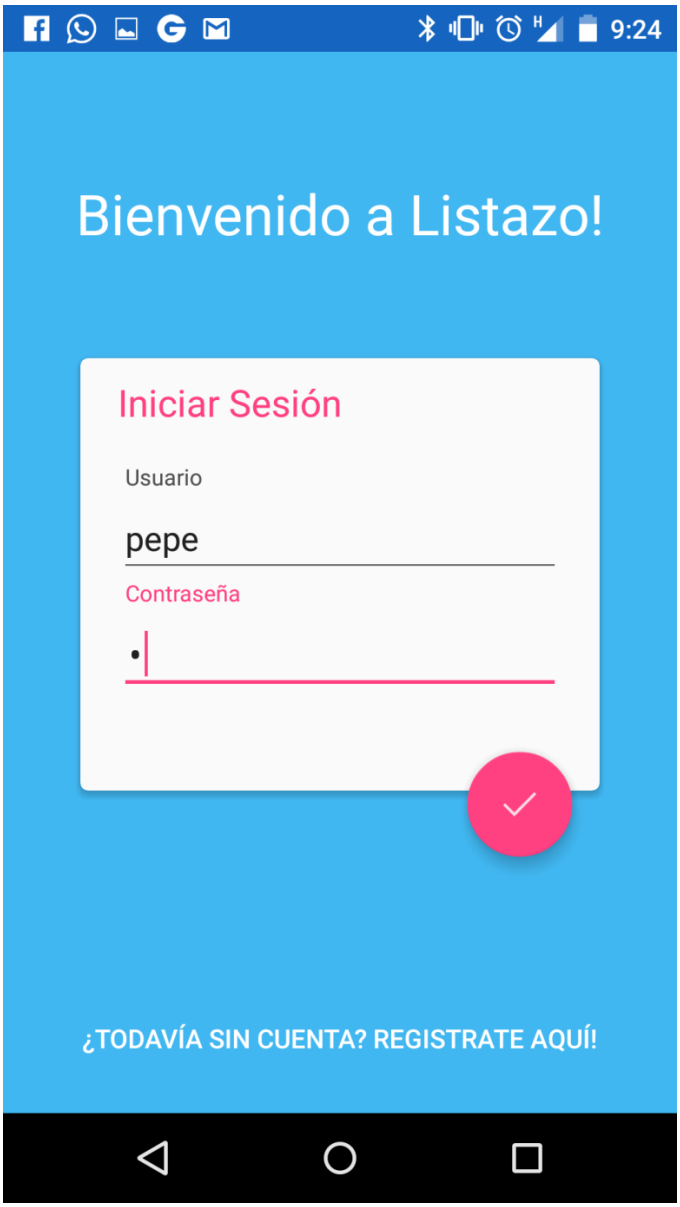
En la foto se puede observar la base de datos de la app, la cual se compone de 7 sencillas tablas relacionadas entre sí.

Descripción de las tablas:

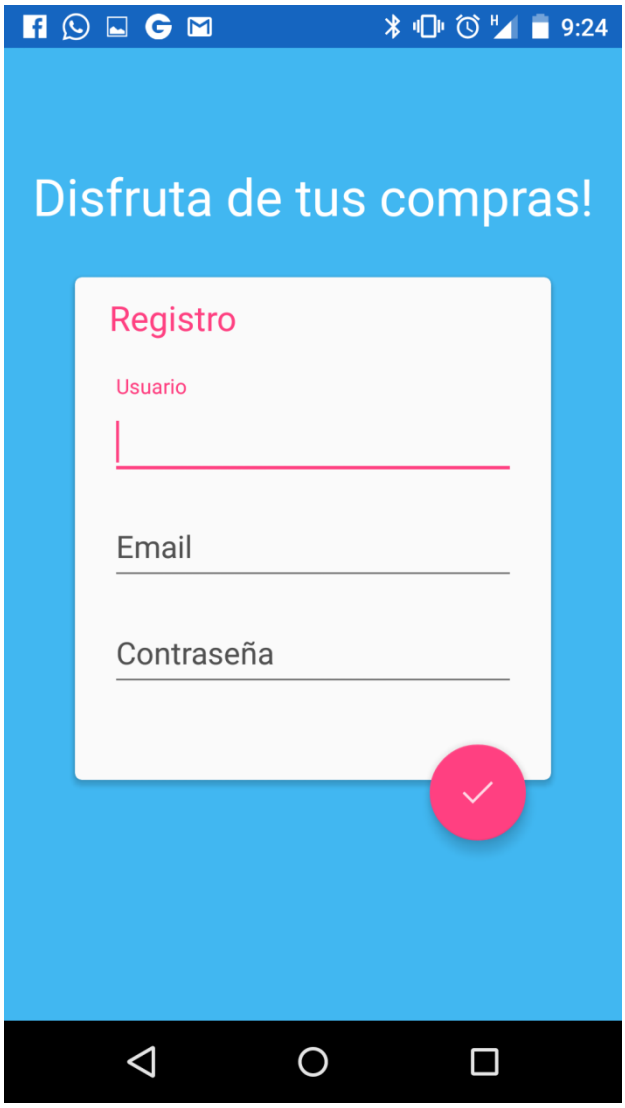
- **User**: Esta tabla almacena los datos de usuario, así como su pushCode o más conocido como Registration ID para el envío de notificaciones Push.
- **FriendShip**: Esta tabla es utilizada para guardar las amistades. Cuando se realiza una petición de amistad, se crea un registro en ella, con el campo confirmed a 0(false). Una vez el otro usuario ha confirmado la amistad, se actualiza el campo confirmed a 1(true) y se guarda la fecha de confirmación.
- Después tenemos las tablas **categories, articles y lists,** que como sus nombres indican, guardan los datos de las categorías, artículos y listas, respectivamente. Estas tablas además están relacionadas con la tabla usuario ya que todo elemento tiene un creador(owner).
- **User_has_list**: Esta tabla sale de la relación entre usuario y listas, ya que un usuario puede ser miembro de varias listas y las listas pueden contener varios usuarios, por lo que necesitamos esta tabla para la relación entre ellos.
- **List_has_article**: Surge de la relación entre listas y artículos, ya que una lista puede contener varios artículos, pero además, esos artículos pueden estar en más de una lista, por lo que necesitamos esta tabla intermedia que las relacione. Cabe destacar su campo <u>enable</u> que indica si el artículo está ya en el carro o no. Poniendo un ejemplo: Conforme vamos comprando los artículos se van tachando de la lista, apareciendo en la parte de abajo, en una nueva categoría llamada "<u>en el carro de la compra</u>" queriendo decir que ya lo hemos seleccionado y está en el carro o cesta de la compra. En ese momento, los artículos pasan a tener el estado enable a 0 (false).

Vistas

Cuando iniciamos la aplicación por primera vez, saldrá la ventana de login

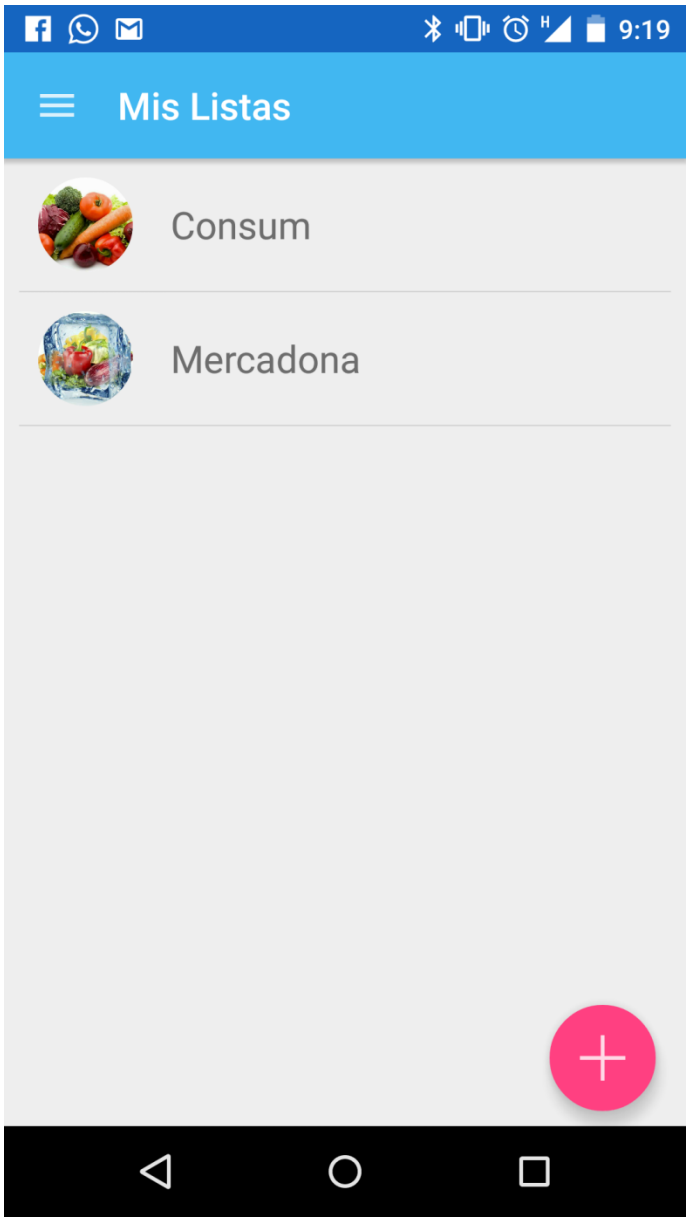


Desde la que podemos iniciar sesión si ya tenemos una cueta o pulsar al enlace de abajo para registrarse, apareciendo la siguiente pantalla de registro:

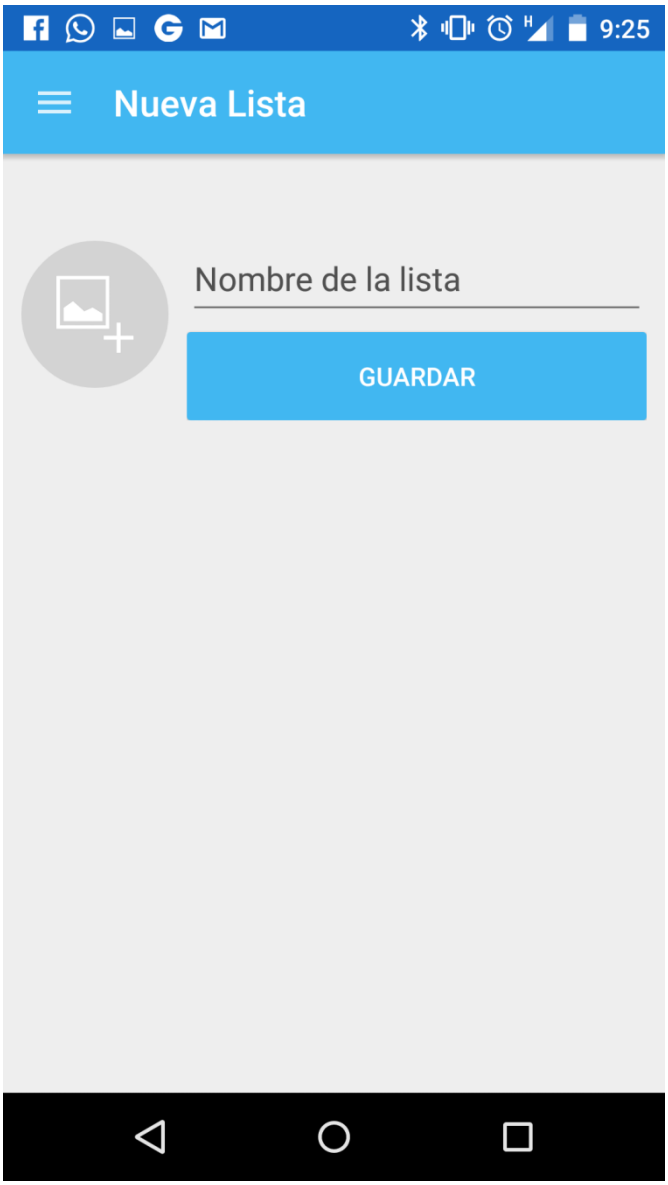


En la que con solo insertando un usuario, email y contraseña, crearemos una cuenta con la que poder disfrutar de listazo.

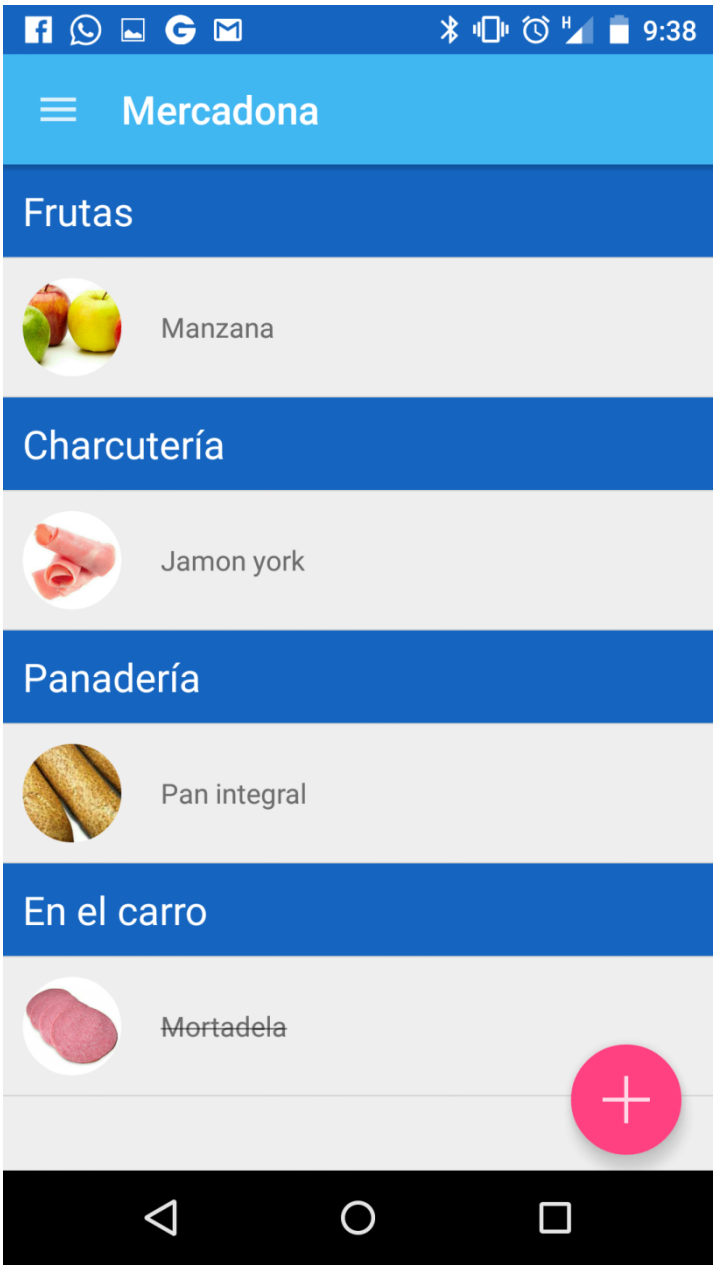
Una vez hemos iniciado sesión, ya sea por login o registro, aparece una ventana principal donde ver nuestras listas, estas pueden ser creadas por nosotros o compartidas por otros usuarios.



Aquí podemos o crear una nueva desde el botón inferior derecho indicando un nombre y foto que la identifique:



O abrir cualquier lista para ver sus artículos y poder gestionarlos:



Fijate como en la parte de abajo vemos la categoría en el carro, con el artículo mortadela tachado. Esto quiere decir que ese artículo ya ha sido seleccionado de su estantería y lo tenemos ya en el carro o cesta de la compra.

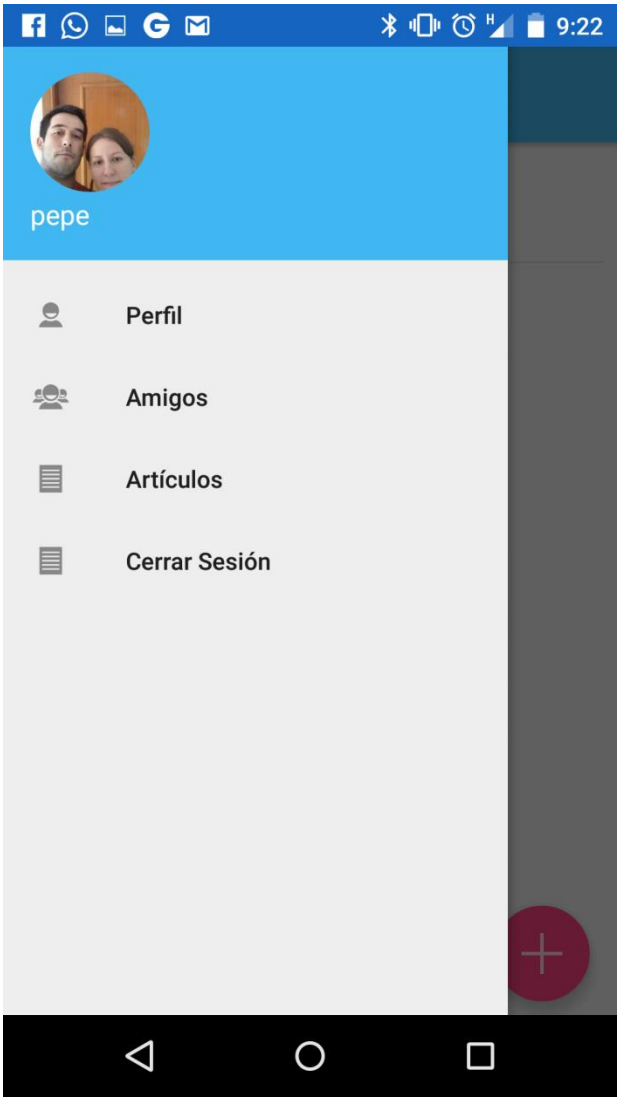
Ahora, de la misma forma que hemos hecho para las listas, podemos añadir artículos con el botón inferior derecho, donde nos aparecerán las distintas categorías:

Por defecto aparecen ya muchas categorías pero siempre podemos añadir de la misma forma que en los demás apartados, por lo que no entrare mas en detalle.

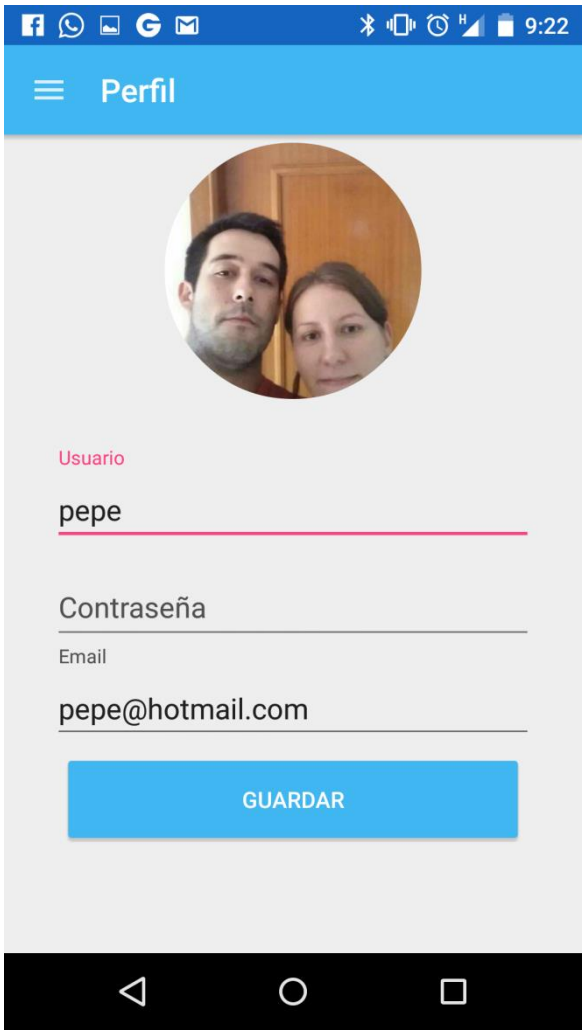
Buscamos la categoría donde está el artículo que deseamos añadir y aparecerán todos los artículos relacionados:

Para añadir el artículo en la lista, tan solo tenemos que pulsar encima de el.

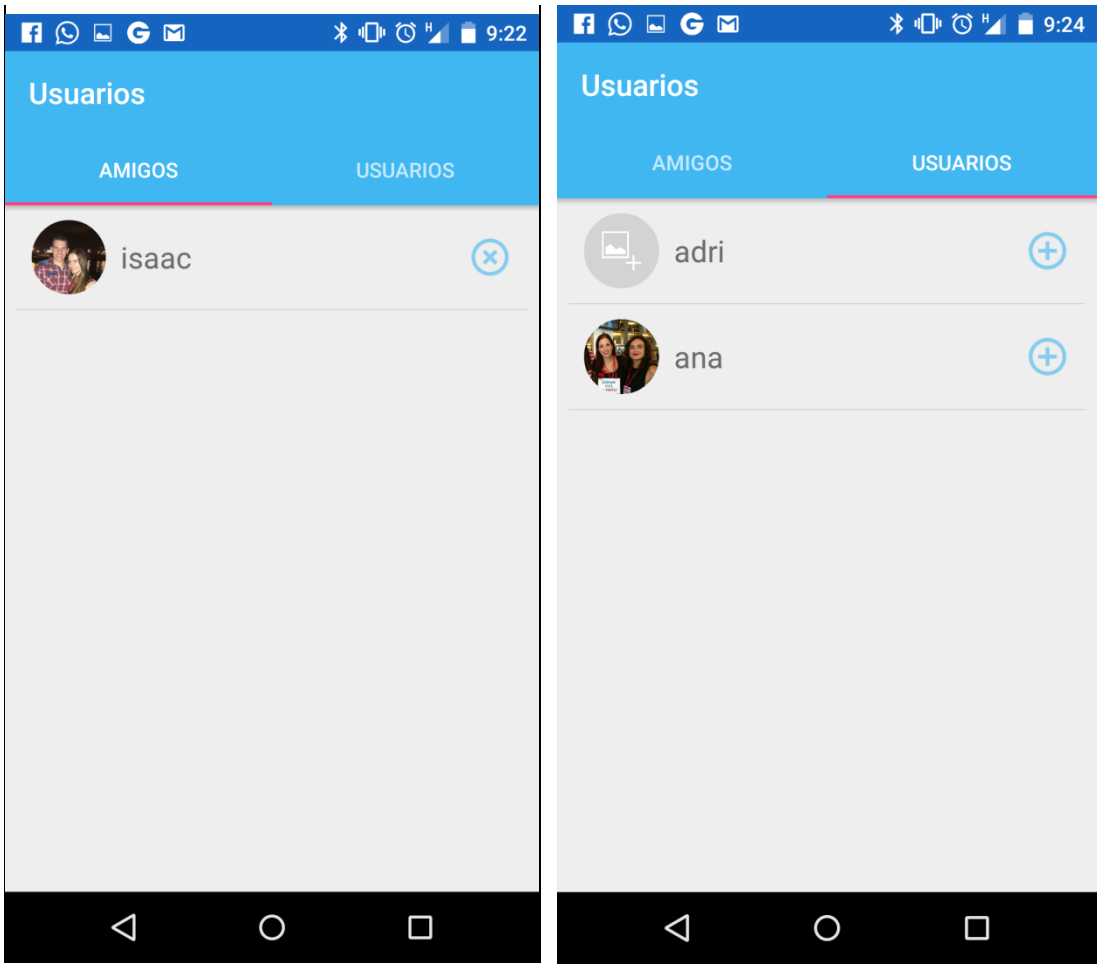
Por otra parte, podemos gestionar nuestro perfil, amigos, categorías y artículos, sin necesidad de entrar en una lista desde la Navigation drawer, pulsando sobre el icono superior izquierdo:



Si pulsamos sobre perfil, podemos actualizar los datos de usuario:

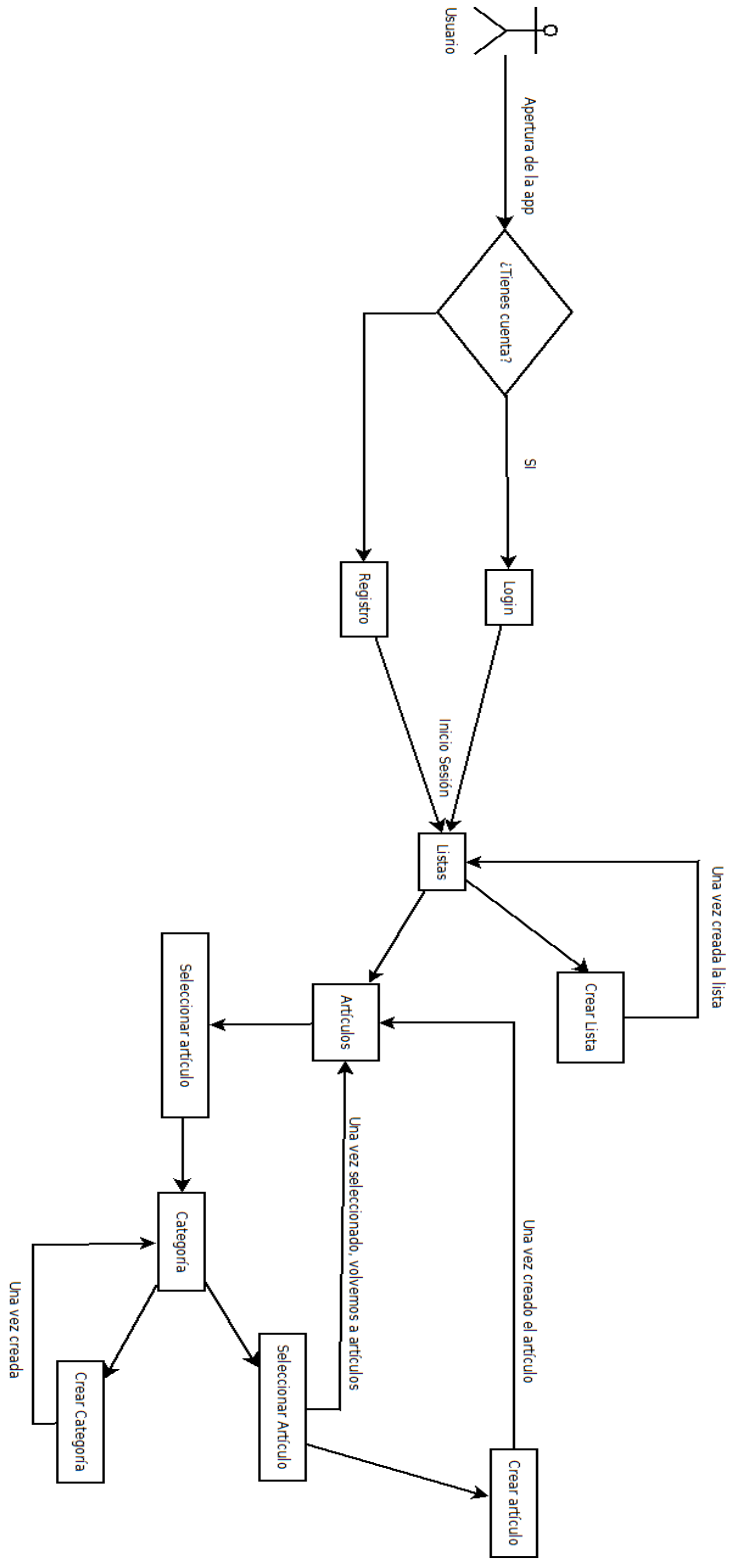


Otro apartado muy importante es el de amigos, ya que solo se podrán compartir las listas con nuestros amigos. Por eso, en este apartado tenemos dos pestañas, una con los amigos y otra con los usuarios de la app para poder enviar peticiones de amistad(Notificaciones push).



Por último comentar, que para compartir una lista tenemos que realizar una pulsación larga sobre encima de ella y pulsar la opción de compartir, seleccionando así los amigos con los que queremos compartirla.

Un diagrama del flujo de la aplicación podría ser el siguiente:



NOTA: Se adjunta además la imagen ya que parece que no se ve muy bien.

Conclusiones

Como conclusiones podemos destacar que:

- **NodeJS**: Aunque es una tecnología que está en auge y cada vez se utiliza más por su rapidez, hay que tener especial cuidado ya que si se produce una excepción, el API cae, dejando de funcionar, por lo que hay que implementar alguna solución para que se restablezca de forma automática y evitar descontentos por caídas del servicio, además de gestionar bien el manejo de excepciones para evitarlas en la medida de lo posible.
- **Swagger:** Este sencillo framework ayuda mucho en la gestión del API así como en la creación de la documentación, por lo que es muy apropiado gastarlo en todo API que se valore.
- **Frames:** He usado al máximo estos componentes para evitar la carga de muchas actividades en la aplicación, aunque a veces cambiar muchas veces de frame sobre la misma actividad, produce que se queden superpuestos al volver atrás. Creo que lo he solucionado en la medida de lo posible pero puede que alguna vez ocurra por lo que tengo que ver esto con más detalle.

Considero las líneas abiertas de ampliar la app a los smartwatchs ya que su uso cada vez está más extendido y es más práctico poder ir viendo la lista desde el reloj, teniendo las manos libres para coger los artículos deseados.

Se ha realizado todo el proyecto con la interfaz material design, ya que es la utilizada hoy en día por google.

Personalmente, me encanta esta interfaz y sus efectos, por lo que he intentado sacar el máximo partido de ellos y de todos sus nuevos componentes como son los nuevos CoordinationLayout, TabLayout, Floating action button…

Anexos

Listado de fuentes entregadas

Más información sobre Swagger: http://swagger.io/

Documentación sobre NodeJS: https://nodejs.org/en/docs/

Capa gratuita de Amazon Web Services: https://aws.amazon.com/es/free/

Material design: https://www.google.com/design/spec/material-design/introduction.html