

Aplicación de solicitudes de recetas y de ingredientes.

Diego Alejandro García Arenas, Oscar Ivan Gomez Zamudio, Yesid Alberto Ochoa Luque, Sergio Ivan Sanchez Plazas
(diagarciaar, oigomezz, yaochoal, sisanchezp)@unal.edu.co

Abstract—Receticapp is an application designed for those who want to cook something different within their menu but they do not do it because they have to go get the ingredients to the store. Receticapp offers to buy the ingredients of a recipe in stores near a click away.

Index Terms—Aplicaciones Móviles, API-REST, Firebase, API Google Maps.

I. INTRODUCCIÓN

LOS amantes de la cocina han querido seguir nuevas recetas, e innovar a partir de ellas, creando otras para mejorar su sazón y explorar nuevos sabores, texturas y demás. Es un reto que están dispuestos a correr cada día. Hasta que ven la lista de ingredientes y surgen los inconvenientes. Entre los cuales se encuentran no tener los insumos para desarrollar la misma, no tener el tiempo para ir por ellos o que la receta no salga como se espera. .

Por eso se planteo el desarrollo de una aplicación compatible con Android, que permitirá, con un solo CLICK, ordenar a partir de una receta básica escogida entre un conjunto de opciones presentadas y un número de porciones deseado. A esta selección responderá la aplicación con una exhibición de los ingredientes necesarios en cantidad, costo y tiempo requeridos para su consecución. En forma completamente interactiva, el usuario seleccionará entre los ingredientes exhibidos los de su interés y las porciones solicitadas de cada uno de ellos. La aplicación modificará los valores de costo y tiempo según lo seleccionado. Una vez aprobada la selección se realizará la compra a los precios señalados y con los proveedores más cercanos. .

II. METODOLOGÍA

Una arquitectura de la aplicación hace uso de una conexión por medio de peticiones a un back-end. En este back se hace una conexión de las recetas con los ingredientes. Una arquitectura que conecta por medio de un api rest a un back-end en donde por medio de peticiones enviadas para tener la solicitud de las recetas como de los ingredientes de estas. Posteriormente mostrara los ingredientes relacionados a esta y estos redireccionararan a la opción de solicitar los mismos a la plataforma .

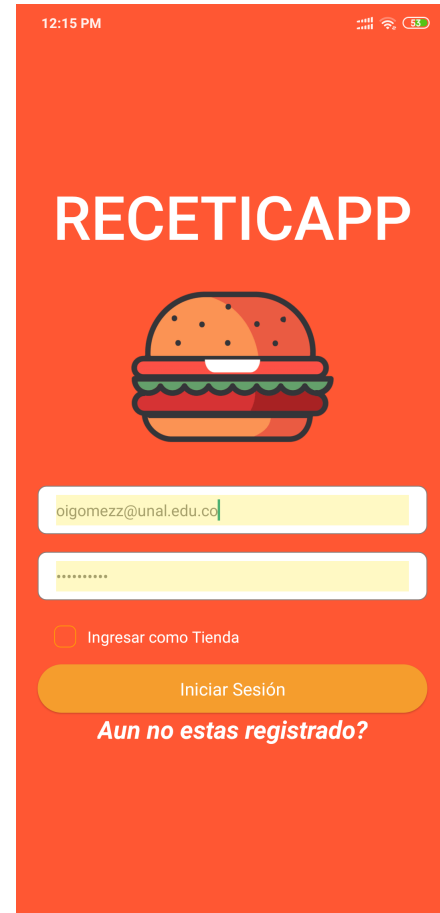


Figura 1. Correo de bienvenida al sistema.

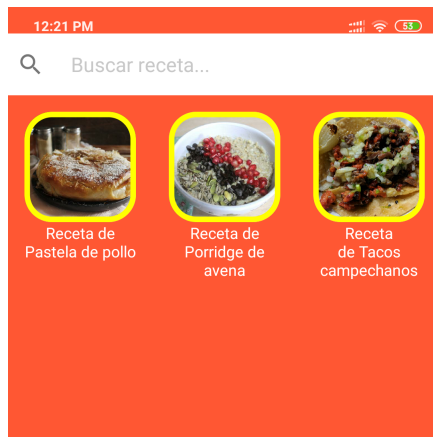


Figura 2. Vista recetas.

Partiendo de esa base, la implementación de métodos de seguridad al mismo convertiría, potencialmente, a un desarrollo elaborado sobre micro-servicios en un desarrollo altamente estable y protegido por lo cual, en nuestro caso, aplicamos al Sistema ReceticApp algunos métodos de seguridad, que describiremos a continuación:

A. Requerimiento funcionales

Entre los requerimientos de la aplicación se encontraron como específicos los siguientes.

- 1) El sistema sera soportado en el SO android.
- 2) Permite registrar usurarios(normal, tienda).
- 3) La aplicación permite hacer inicio de sesión con dos tipos distintos de roles.
- 4) Los usuarios podrán subir imágenes y personalizar su cuenta
- 5) Las tiendas podrán subir sus recetas .
- 6) Los usuarios podrán hacer pedidos de los ingredientes. puntuales de la receta.
- 7) La aplicación puede visualizar las tiendas cercanas que estén según GPS.
- 8) La aplicación calcula la cantidad de ingredientes según numero de platos.
- 9) Los usuarios podrán calificar la orden una vez se entregue.
- 10) La tienda puede modificar el estado de los pedidos.

B. Requerimiento no funcionales

Los requerimientos no funcionales que se hallaron fueron (Reciliencia a ataques DDOS.

- 1) Escalabilidad de usuarios.
- 2) Se utilizara la API de sistema de pagos(pay-U).
- 3) Se utilizara el sistema de mapas traído de google.
- 4) Se utiliza el sistema de Firebase para guardar e ingresar usuarios.

III. TECNOLOGÍAS

Inicialmente, debido a las especificaciones del proyecto, se vio la necesidad de implementar un mínimo de dos roles asignados al sistema, en donde cada rol tendría la capacidad de hacer búsquedas.

A. Firebase

Se implementó una base de datos y un método de ingreso que aplicaba el servicio de Firebase de esta manera se encontraría segura y permitiría. La mayoría de las apps necesitan identificar a los usuarios. Conocer la identidad de un usuario permite que una app guarde sus datos en la nube de forma segura y proporcione la misma experiencia personalizada en todos los dispositivos del usuario.

Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en la aplicación. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y mucho más.

Firebase Authentication se integra estrechamente en otros servicios de Firebase y aprovecha los estándares de la industria como OAuth 2.0 y OpenID Connect, por lo que se puede integrar fácilmente con un backend personalizado. Para que un usuario acceda a la app, primero se debe obtener credenciales de autenticación del usuario. Estas credenciales pueden ser la dirección de correo electrónico y la contraseña del usuario, o un token OAuth de un proveedor de identidad federada. A continuación, debes pasar estas credenciales al SDK de Firebase Authentication. Luego, el servicio de backend verificarán esas credenciales y mostrarán una respuesta al cliente.

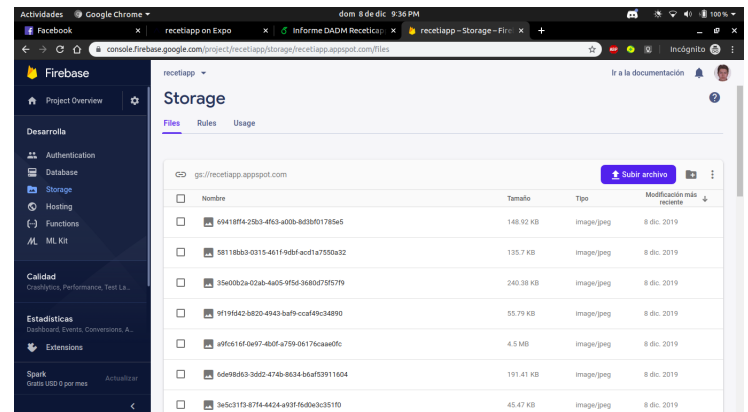


Figura 3. Instancias de Firebase.

B. Ruby on Rails

Se uso la parte de back-end sobre este framework dada su facilidad para desarrollar modulos correspondientes a los pedidos o y a la creacion de recetas. Adicionalmente se encuentra la capacidad de hacer conexiones por medio de API-REST a el front. Cabe resaltar que la para la parte de la creación se uso la facilidad de el uso de este framework con PostgreSQL.

C. React Native

React es una biblioteca Javascript para crear interfaces de usuario interactivas de forma sencilla. Diseña vistas simples para cada estado de la aplicación, y React se encargará de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien. React ha sido diseñado

desde su inicio para ser adoptado gradualmente, así puedes usar tan poco o mucho de React como se necesite.

A medida que la aplicación crezca, se puede considerar una configuración más integrada. Hay una gran variedad de herramientas que recomendamos para aplicaciones más grandes. Cada una de ellas puede funcionar con poca configuración, y permite tomar toda la ventaja de un ambiente React completo. Las vistas declarativas hacen que el código sea más predecible, por lo tanto, fácil de depurar.

D. Expo

Expo es un conjunto de herramientas creadas en torno a React Native y, aunque tiene muchas características, la característica más relevante para nosotros en este momento es que puede hacer que escriba una aplicación React Native en cuestión de minutos. Solo necesitará una versión reciente de Node.js y un teléfono o emulador.

Estas herramientas lo ayudan a comenzar rápidamente, pero antes de comprometerse a crear su aplicación con Expo CLI. Expo CLI le permite ejecutar su aplicación React Native en un dispositivo físico sin configurar un entorno de desarrollo.

Debido a que no construye ningún código nativo cuando usa Expo para crear un proyecto, no es posible incluir módulos nativos personalizados más allá de React Native API y componentes que están disponibles en la aplicación cliente de Expo.

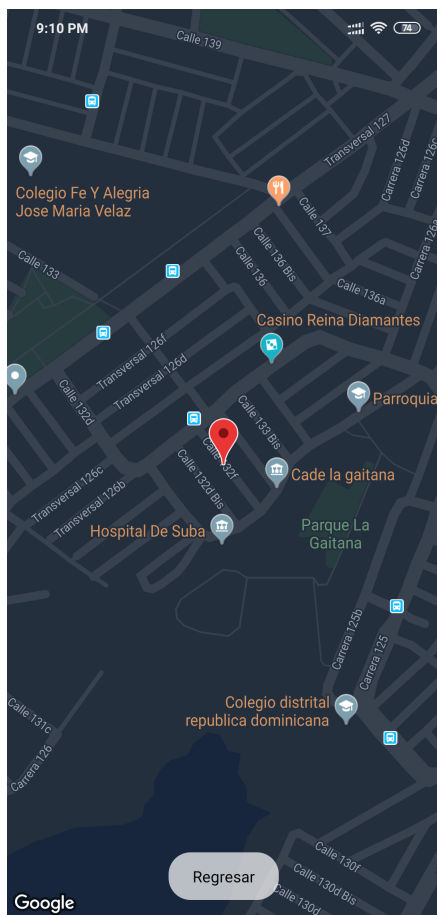


Figura 4. Solicitud en el mapa.

IV. CONCLUSIONES

- Al hacer el desarrollo de la aplicación se puede notar en retrospectiva que el nicho de mercado aun puede ser explotado, por lo que una aplicación de este estilo es bien recibida por el público.
- Dado que la plataforma de prueba se encuentra en la nube, con el uso de la consola se simuló el resultado de un ataque de inyección SQL, que eventualmente podría solicitar la eliminación de una tabla y se confirmó que el sistema no perdió funcionalidad, aunque significó un tiempo adicional reducido destinado a la replicación de una base de datos no afectada (backup).
- Al tener esta aplicación montada en la nube en la plataforma de Heroku se tiene la seguridad de una resiliencia a los ataques DDOS e incluso se puede en planes futuros hacer paso a montar la misma en máquinas de la nube como lo que son Google o AWS ofreciendo Infrastructure as a Service.
- Se sometió a prueba el sistema sometándolo a un ataque DDoS, cuya respuesta, en principio ralentizada como resultado del ataque, puso en ejecución la desviación provista por el proxy inverso, y permitió la correcta atención a los usuarios regulares con una muy pronta recuperación de los estándares de atención.

V. REPOSITORIOS

A. Receticapp Código Fuente

<https://github.com/Receticapp/Receticapp>

B. Receticapp App Online

<https://expo.io/@yaochoal/receticapp>

REFERENCIAS

- [1] R.A. Kemmerer. *Cybersecurity*.
- [2] G.Balaji, V.S.Hari Prassath, V.Sriram(2018). ISSUES BASED ON CYBER CRIME AND SECURITY.
- [3] Firebase *Firestore* from <https://firebase.google.com/docs/auth/>.
- [4] React Native *React Native* from <https://facebook.github.io/react-native/docs/tutorial>
- [5] Expo *Expo* from <https://facebook.github.io/react-native/docs/getting-started>
- [6] R.A. Kemmerer. *Cybersecurity*.