Diseño e implementación de una aplicación móvil android para el seguimiento de rutas de transporte urbano en el municipio de Yopal

Design and implementation of an android mobile application to follow the routes of urban transport in the area of Yopal

Policarpo Malabar Galindo Pérez
Fundación Universitaria de San Gil, UNISANGIL El Yopal, Colombia

Correo electronico: policarpogalindo@unisangil.edu.co

Mónica Andrea Suárez Vargas
Autor de correspondencia
Fundación Universitaria de San Gil, UNISANGIL El Yopal, Colombia

Correo electronico: msuarez@unisangil.edu.co

Información del artículo: recibido: 19 de Abril de 2017, aceptado: 16 de Agosto de 2017
https://10.17081/invinno.5.2.2759

Resumen

Este documento presenta los resultados del desarrollo de una aplicación móvil Android para la visualización de las rutas de transporte urbano en la ciudad de Yopal, utilizando la metodología XP, que ofrece un marco adaptable a las condiciones del proyecto. El desarrollo de la aplicación se ha realizado con el entorno de desarrollo de Android Studio, Google Maps para Android y Firebase para iniciar sesión con la cuenta de Google y la base de datos. El trabajo está dirigido a los usuarios de la ciudad de Yopal, departamento de Casanare (Colombia), que toman el servicio de transporte urbano con el objetivo de facilitar la información correspondiente a las rutas de los paseos en minibús.

Palabras Claves:

Móvil, Android, rutas, usuario

Abstract

This document presents the results of the development of an Android mobile application for the visualization of the urban transportation routes in Yopal city, making use of the agile XP methodology that offers an adaptable framework to the project conditions. The development of the application has been made with the Android Studio development environment, Google Maps for Android, Firebase to login in with the Google account and database. The application is addressed to the users of Yopal City, Casanare department, who take the urban transportation service, to facilitate the information corresponding to the routes the minibus rides.

Keywords:

Mobile, android, routes, user.

La globalización tecnológica ha obligado a las empresas a cambiar su forma de realizar los negocios; debido a la gran competencia que esta ha generado en el mercado. Por esta razón, es necesario que las empresas diseñen herramientas para agilizar los procesos que realizan sin perder calidad.

Por otra parte, ahora las personas tienen al alcance dispositivos móviles que soportan la instalación de aplicaciones y datos o conexión a wi-fi para la navegación en Internet, lo que ha modificado sus hábitos de consumo y la forma de adquirir productos [1].

En estas circunstancias, es importante adoptar los cambios que la tecnología ofrece, para tener una mayor interacción con el usuario. En concreto, el uso de aplicaciones móviles facilita la comunicación entre las empresas de transporte urbano y el usuario, por lo que es conveniente mostrar información de manera rápida y sencilla, de modo que el usuario puede acceder a esta desde sus dispositivos electrónicos.

Los usuarios que usan el servicio de transporte urbano desconocen en muchas ocasiones el trayecto y los lugares por los que pasa el microbús, viéndose con frecuencia en la necesidad de preguntarle sobre esto al conductor, y no pocas veces este detiene el vehículo para responder a la inquietud del usuario.

Analizando esta situación, se plantea como solución el desarrollo de Rutas Yopal, una aplicación móvil en Android, que contará con las rutas actualmente disponibles, además de opciones para que el usuario busque la ruta que mejor le sirve para llegar a su destino.

Durante el desarrollo del proyecto se plantean los requisitos funcionales y no funcionales que la aplicación debe cumplir para su correcto funcionamiento. Estos son necesarios para la realización de los diagramas en UML, los cuales permitirán entender la interacción entre esta y el usuario.

Con los diagramas obtenidos se procede a realizar la codificación haciendo uso del entorno de desarrollo Android Studio. Adicionalmente, se contará con la interacción de la plataforma Firebase para el almacenamiento de los datos a usar en la lista de barrios que contiene la aplicación, y los permisos de Google Maps para que se pueda visualizar el mapa.

La metodología XP (Extreme Programming) es "un enfoque para el desarrollo de software que utiliza buenas prácticas de desarrollo y las lleva a los extremos. Se basa en valores, principios y prácticas esenciales. Los cuatro valores son la comunicación, la simplicidad, la retroalimentación y la valentía" [2].

La metodología XP propone un lineamiento para garantizar que se hagan las cosas que se consideran más relevantes en el proyecto, dividiéndolo en cuatro fases con el fin de no realizar acciones innecesarias:

- Fase de planeación
- Fase de diseño
- Fase de codificación
- Fase de pruebas

Fase de planeación

En la fase de planeación se llevaron a cabo reuniones con el jefe operativo de la empresa Coocatrans Ltda., para conocer el número de rutas existentes y de esta forma iniciar la creación de las historias de usuario y definir los requerimientos funcionales y no funcionales para el desarrollo de la aplicación. A continuación, se explica qué significan estas últimas:

Historias de usuario: Constituyen la representación de un requisito que se escribe usando un lenguaje común con el usuario, para determinar las funciones que debe cumplir la aplicación.

Historia de usuario	
Numero: 01	Usuario: Cliente
Nombre historia: Crear rutas de transporte urbano	
Prioridad en negocio: Alta	Riesgo de desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 01
Programador responsable: Policarpo Galindo	
Descripción: Trazar las rutas que realizan las busetas en el área urbana de Yopal.	
Validación: El usuario puede seleccionar una de las rutas establecidas y ver el trayecto.	

Tabla 1.
Crear rutas de transporte.

Historia de usuario	
Numero: 02	Usuario: Cliente
Nombre historia: Crear buscador de ruta	
Prioridad en negocio: Alta	Riesgo de desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 02
Programador responsable: Policarpo Galindo	
Descripción: Crear una lista de barrios por los cuales pasa la buseta, para poder consultar escribiendo el nombre del barrio.	
Validación: El usuario obtendrá una respuesta luego de escribir el nombre de un barrio.	

Tabla 2.
Crear bucador de rutas

Requerimientos funcionales: Indican cómo debe responder la aplicación cuando el usuario interactúa con ella.

Tabla 3.
Autenticación de usuario

Código	RF01	Prioridad	E-Esencial
Título		Autenticación de usuario	
Descripción:			
El usuario deberá autenticarse utilizando una cuenta activa de Google para acceder a la aplicación. Una vez autenticado no tendrá que realizar este proceso hasta que seleccione la opción "cerrar sesión". Si el usuario no se ha autenticado, tendrá la opción de añadir una cuenta para que seleccione una existente o cree una nueva cuenta en Google.			

Tabla 4.
Localización

Código	RF02	Prioridad	E-Esencial
Título		Localización	
Descripción:			
Al ingresar el usuario a la aplicación, está lo georreferenciará en el punto de localización en el que se encuentre respecto a las coordenadas del mapa de Yopal.			

Tabla 5.
Búsqueda de barrios

Código	RF03	Prioridad	E-Esencial
Título		Búsqueda de barrios	
Descripción:			
El usuario tendrá la opción de hacer una búsqueda ingresando el nombre del barrio y la aplicación mostrara las rutas que pasan por este barrio. Esta información se mostrará en una lista.			

Código	RF04	Prioridad	E-Esencial
Título		Trayecto	
Descripción:			
El usuario seleccionara una de las opciones que se encuentra en el menú (ruta 1, …ruta 13) y la aplicación le mostrará en un mapa el trayecto que sigue la buseta.			

Tabla 6.
trayecto

Código	RF05	Prioridad	E-Esencial
Título		Cerrar sesión	
Descripción:			
El usuario podrá salir de la aplicación seleccionando la opción "Cerrar sesión" que se encuentra en el menú.			

Tabla 6.
Cerrar sesión

Requerimientos no funcionales: Son necesarios para el correcto funcionamiento de la aplicación.

Tabla 8.
Facilidad de uso

Código	RF01	Prioridad	E-Esencial
Título		Facilidad de uso	
Descripción:			
ELa aplicación debe ser fácil de usar, para todo tipo de usuarios (expertos y no expertos).			

Tabla 9.
Dispositivos Android

Código	RF02	Prioridad	E-Esencial
Título		Dispositivos Android	
Descripción:			
La aplicación se ejecutará en dispositivos Android versión 4.2.2 y posteriores.			

Tabla 10.
Base de datos Firebase

Tarea de ingeniería	
Número tarea: 2	Número historia: 1. Crear rutas de transporte urbano
Nombre tarea: Visualizar el trayecto de las rutas	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha de inicio: Agosto 16 de 2016	Fecha Fin: Octubre 15 de 2016
Programador responsable: Policarpo Malabar Galindo Pérez	
Descripción: Desarrollar y programar la interfaz para que los usuarios puedan visualizar el trayecto que realiza la buseta, a través de la aplicación.	
Validación: El usuario obtendrá una respuesta luego de escribir el nombre de un barrio.	

Código	RF03	Prioridad	E-Esencial
Título		Base de datos Firebase	
Descripción:			
Se hace uso de la plataforma Firebase para almacenar la información de los barrios.			

Código	RF04	Prioridad	E-Esencial
Título		Google Maps	
Descripción:			
Habilitar en la consola de desarrollador de Google los permisos para hacer uso de Google Maps.			

Tabla 11.
Google Maps

Código	RF05	Prioridad	E-Esencial
Título		Autenticación Firebase	
Descripción:			
Habilitar en la plataforma de Firebase el inicio de sesión con cuenta de Google.			

Tabla 12.
Autenticación Firebase

Tabla 13.
Conexión a internet

Código	RF06	Prioridad	E-Esencial
Título		Conexión a internet	
Descripción:			
Se requiere que el dispositivo cuente con conexión a Internet para acceder a la información que la aplicación contiene.			

Tareas de ingeniería

Tabla 14.
Diseño de diagramas UML

Tarea de ingeniería	
Número tarea: 1	Número historia:
Nombre tarea: Diseño de diagramas UML	
Tipo de tarea: Diseño	Puntos estimados:
Fecha de inicio: Julio 29 de 2016	Fecha Fin: Agosto 15 de 2016
Programador responsable: Policarpo Malabar Galindo Pérez	
Descripción: Realizar diagramas UML (Diagrama de casos de uso, diagrama de actividades y diagrama de secuencias), para comprender la interacción entre el usuario y el sistema.	
Validación: El usuario obtendrá una respuesta luego de escribir el nombre de un barrio.	

Tarea de ingeniería	
Número tarea: 3	Número historia: 2. Crear buscador de ruta
Nombre tarea: Buscar y mostrar barrios	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha de inicio: Noviembre 15 de 2016	Fecha Fin: Diciembre 14 de 2016
Programador responsable: Policarpo Malabar Galindo Pérez	
Descripción: Se creará una lista de barrios, para que los usuarios puedan consultar la ruta que quieren, escribiendo el nombre del barrio.	
Validación: El usuario obtendrá una respuesta luego de escribir el nombre de un barrio.	

Tabla 15.
Buscar y mostrar barrios

Tarea de ingeniería	
Número tarea: 4	Número historia:
Nombre tarea: Inicio de sesión	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha de inicio: Diciembre 14 de 2016	Fecha Fin: Enero 16 de 2017
Programador responsable: Policarpo Malabar Galindo Pérez	
Descripción: Desarrollar y programar la interfaz de inicio de sesión, para que los usuarios ingreses los datos (correo y contraseña), que serán validados por el sistema, para realizar el registro en la aplicación. Una vez validado el usuario ingresara a la actividad principal de la aplicación.	
Validación: El usuario obtendrá una respuesta luego de escribir el nombre de un barrio.	

Tabla 16.
Inicio de sesión

Fase de diseño

En la fase de diseño se realizan los diagramas UML (Lenguaje Unificado de Modelado), a fin de tener una perspectiva de lo que hace el sistema al interactuar con el usuario.

Casos de uso: Se establecen los actores que van a intervenir en el desarrollo de la aplicación, teniendo en cuenta los requisitos planteados anteriormente, para dar cumplimiento a la tarea que cada uno debe cumplir en la la realización del proyecto [3].

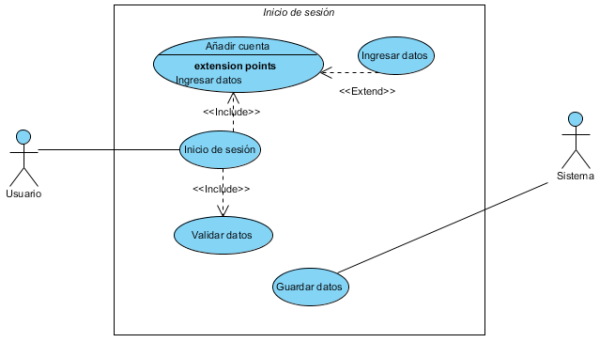


Figura 1.
Caso de Uso - Inicio de sesión

Nombre del caso de uso:	Inicio de sesión
Actor:	Usuario
Propósito:	Ingresar a la aplicación
Resumen:	El usuario ingresa sus datos de su cuenta en Google (correo y contraseña).
Pre-condición	El usuario debe tener Internet (datos o wi-fi).
Usuario	Sistema
1. El usuario ingresa a la aplicación	2. El sistema muestra botón de iniciar sesión
3. El usuario selecciona botón e ingresa los datos correspondientes	4. El sistema valida la información suministrada. Si los datos son válidos los envía a la actividad principal; de lo contrario, envía un mensaje informando que no se ha podido iniciar sesión.
	5. El sitema guarda la información.

Tabla 17.
Caso de uso - inicio de sesión

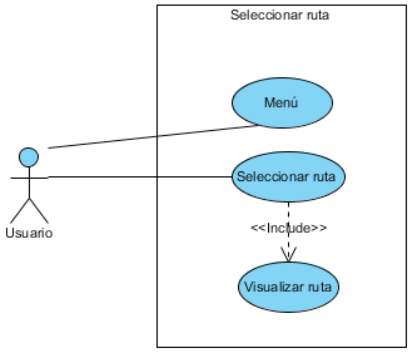


Figura 2.
Caso de Uso – Seleccionar ruta

Nombre del caso de uso:	Seleccionar ruta
Actor:	Usuario
Propósito:	Seleccionar una opción del menú
Resumen:	Cuando el usuario haya iniciado sesión, podrá seleccionar una ruta y ver el trayecto trazado en el mapa.
Pre-condición	El usuario debe tener internet (datos o wi-fi).
El usuario debe haber iniciado sesión.	Sistema
Usuario	Sistema
1. El usuario inicia sesión	2. El sistema muestra el mapa
3. El usuario selecciona una ruta	4. El sistema muestra el trayecto

Tabla 18.
Caso de uso – seleccionar ruta

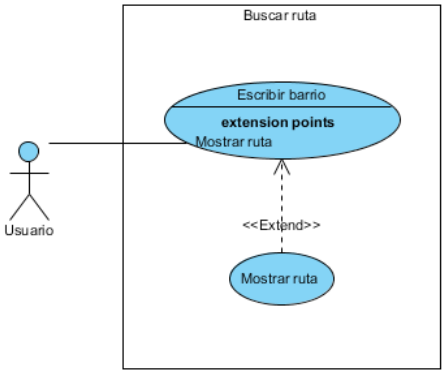


Figura 3.
Caso de Uso – Buscar ruta

Nombre del caso de uso:	Buscar ruta
Actor:	Usuario
Propósito:	Filtrar lista de acuerdo con el barrio que se escriba
Resumen:	Cuando el usuario seleccione la opción de barrios, se mostrará la lista de barrios y contará con la opción de búsqueda para filtrar los resultados.
Pre-condición	El usuario debe tener Internet (datos o wi-fi).
El usuario debe haber iniciado sesión.	Sistema
Usuario	Sistema
1. El usuario selecciona opción barrios	2. El sistema muestra lista de barrios
3. El usuario escribe el nombre de un barrio	4. El sistema filtra la lista y muestra los resultados que corresponden con la búsqueda.

Tabla 19.
Caso de uso - buscar ruta

Diagramas de secuencias: Muestra una interacción que representa la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores. El tiempo fluye por el diagrama y muestra el flujo de control de un participante a otro. Se utilizan diagramas de secuencia para visualizar instancias y eventos en lugar de clases y métodos. En el diagrama, puede aparecer más de una instancia del mismo tipo. También puede haber más de una ocurrencia del mismo mensaje. [4].

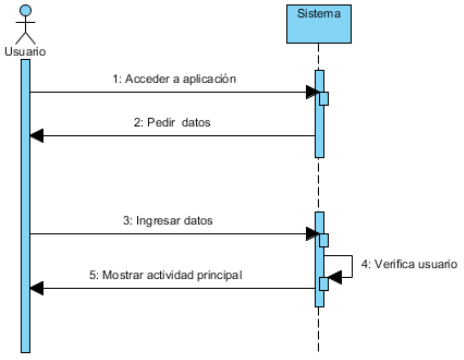


Figura 4.
Diagrama de secuencia - Inicio de sesión

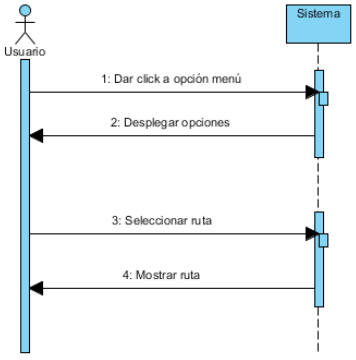


Figura 5.
Diagrama de secuencia – Seleccionar ruta

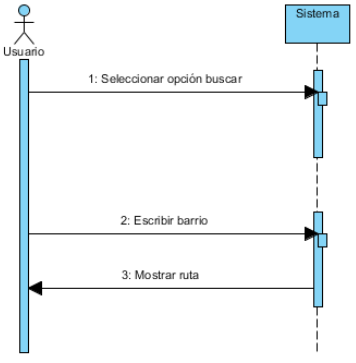


Figura 6.
Diagrama de secuencia – Buscar ruta

Diagramas de actividades: Muestra el proceso que realiza el programa a través de una serie de acciones entre el usuario y el sistema al momento de ejecutarlo [5].

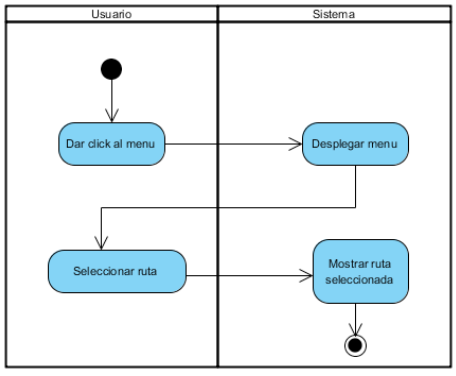


Figura 7.
Diagrama de actividades - Inicio de sesión

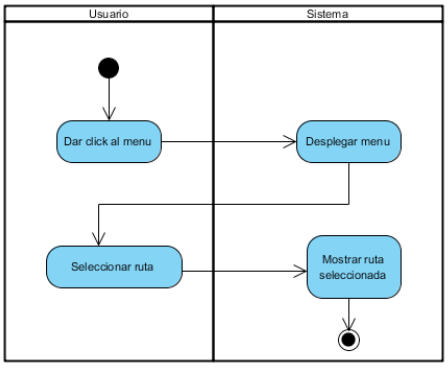


Figura 8.
Diagrama de actividades – Seleccionar ruta

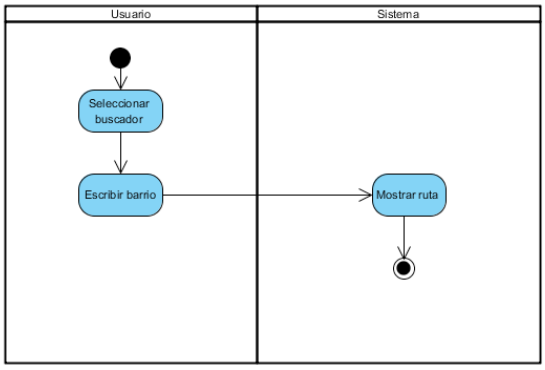


Figura 9.
Diagrama de actividades – Buscar ruta

Fase de codificación

Se mostrará una parte del código y la interfaz gráfica que se utilizó para llevar a cabo el proyecto.

Login: Se habilita en la plataforma de Firebase el uso de la cuenta de Google para el inicio de sesión (ver Fig. 10).

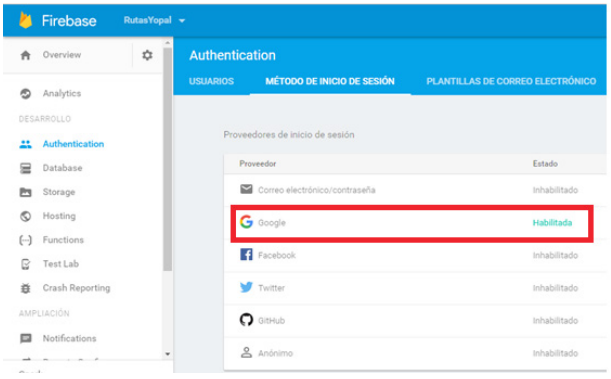
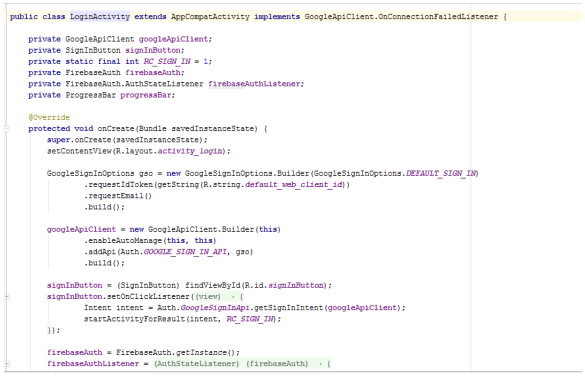


Figura 10.
Método de inicio de sesión

En la Fig. 11, por otro lado, se enseña parte del código que se usó para llevar a cabo la comprobación y el registro de la cuenta de Google en la aplicación.



```
public class LoginActivity extends AppCompatActivity implements GoogleApiClient.OnConnectionFailedListener {

    private GoogleApiClient googleApiClient;
    private SignInButton signInButton;
    private static final int RC_SIGN_IN = 1;
    private FirebaseAuth firebaseAuth;
    private FirebaseAuth.AuthStateListener firebaseAuthListener;
    private ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();

        googleApiClient = new GoogleApiClient.Builder(this)
                .enableAutoManage(this, this)
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();

        signInButton = (SignInButton) findViewById(R.id.signInButton);
        signInButton.setOnClickListener((view) -> {
                Intent intent = Auth.GoogleSignInApi.getSignInIntent(googleApiClient);
                startActivityForResult(intent, RC_SIGN_IN);
        });

        firebaseAuth = FirebaseAuth.getInstance();
        firebaseAuthListener = (AuthStateListener) (firebaseAuth) -> {
```

Figura 11.
Código para LoginActivity

La Fig. 12 constituye la interfaz necesaria para que el usuario seleccione una cuenta o añada otra para ingresar a la información de la aplicación.

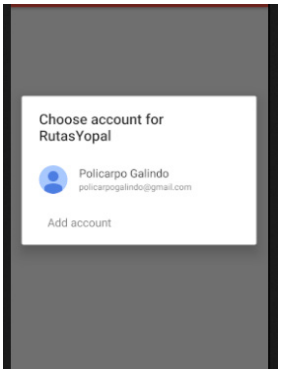


Figura 11.
Interfaz de inicio de sesión

Menú: Para la navegación en la aplicación se ha usado un NavDrawer, que permite tener un menú desplegable, para que el usuario seleccione una de las opciones.

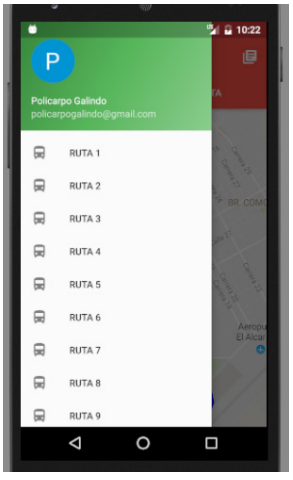
En la Fig. 13 se enseña parte del código que se usó para crear el menú en la aplicación.

Figura 13.
Código para menú

```
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu, menu);
    menu.findItem(R.id.search).setVisible(false);
    menu.findItem(R.id.lista_barrios).setVisible(true);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.

    fragmentManager = getSupportFragmentManager();

    switch (item.getItemId()) {
        case R.id.lista_barrios:
            fragment = new SearchFragment();
            break;
```

La Fig. 14 reproduce la interfaz para que el usuario seleccione una de las opciones de ruta para visualizarla en el mapa

Figura 14.
Interfaz del menú



Google Maps: En la consola se debe registrar el desarrollador de Google, para habilitar la Api de Google Maps y tener el permiso de usarla en la aplicación móvil (ver Fig. 15).

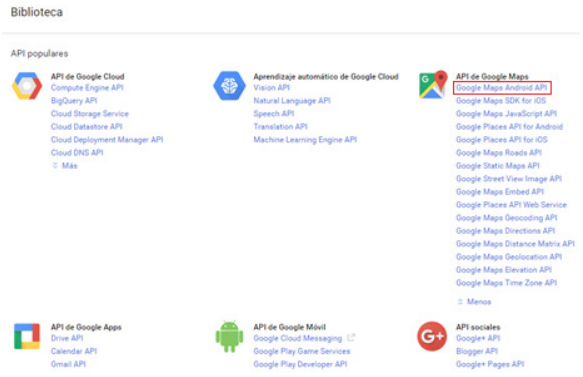
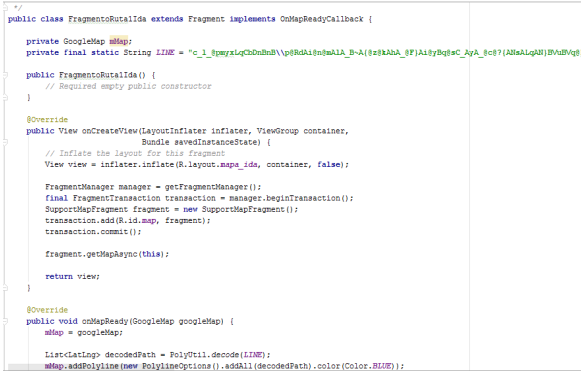


Figura 15.
Habilitar Google Maps

En la Fig. 16 se enseña parte del código que se usó para implementar el método OnMapReadyCallback, para que la poli línea se pueda ver en el mapa al momento de ejecutar la aplicación.



```
*/
public class FragmentoRutaIda extends Fragment implements OnMapReadyCallback {

    private GoogleMap mMap;
    private final static String LINE = "c_l_@pmyxLqCbDnBnB\\p@RdAi@n@mAlA_B~A(@z@kAbA_@F)Ai@yBq@sC_AyA_@c@?(AHsALgAH)BVnBVq@

    public FragmentoRutaIda() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.mapa_ida, container, false);

        FragmentManager manager = getFragmentManager();
        final FragmentTransaction transaction = manager.beginTransaction();
        SupportMapFragment fragment = new SupportMapFragment();
        transaction.add(R.id.map, fragment);
        transaction.commit();

        fragment.getMapAsync(this);

        return view;
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        List<LatLng> decodedPath = PolyUtil.decode(LINE);
        mMap.addPolyline(new PolylineOptions().addAll(decodedPath).color(Color.BLUE));
```

Figura 16.
Código de fragment

La Fig. 17 es la interfaz de la aplicación para visualizar el trayecto de la ruta en el mapa.

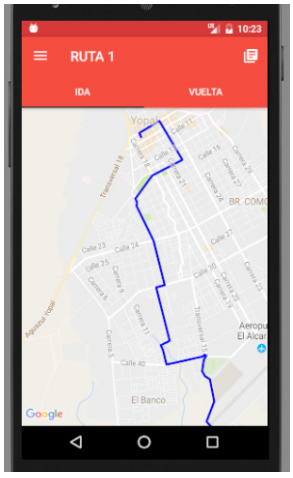


Figura 17.
Interfaz de trayecto

RecyclerView: Esta aplicación se usa para mostrar la información de los barrios en una lista, también se usará para que al momento de hacer la búsqueda se filtre la información y esta se muestra de manera ordenada en la lista.

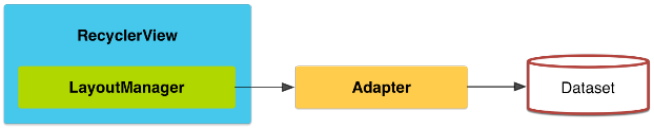
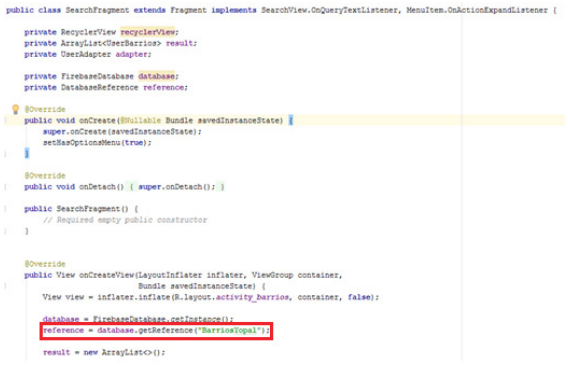


Figura 18.
RecyclerView [6]

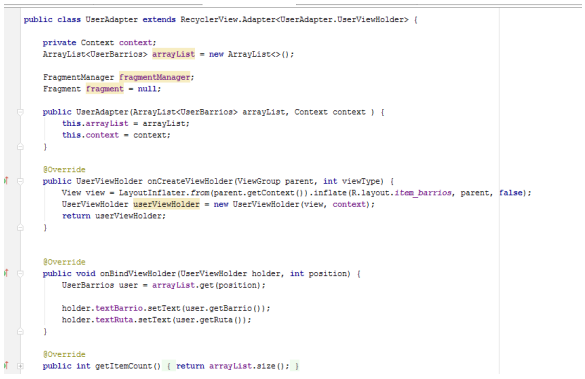
En la Fig. 19 se hace referencia a la conexión con la base de datos en Firebase para que se muestre la información almacenada.



```
public class SearchFragment extends Fragment implements SearchView.OnQueryTextListener, MenuItem.OnActionExpandListener {

    private RecyclerView recyclerView;
    private ArrayList<UserBarrios> result;
    private UserAdapter adapter;

    private FirebaseDatabase database;
    private DatabaseReference reference;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setHasOptionsMenu(true);
    }

    @Override
    public void onDetach() { super.onDetach(); }

    public SearchFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.activity_barrios, container, false);

        database = FirebaseDatabase.getInstance();
        reference = database.getReference("BarriosYopal");

        result = new ArrayList<>();
```

Figura 19.
Código conexión con Firebase

En la Fig. 20 se enseña parte del código del adaptador del RecyclerView.



```
public class UserAdapter extends RecyclerView.Adapter<UserAdapter.UserViewHolder> {

    private Context context;
    ArrayList<UserBarrios> arrayList = new ArrayList<>();

    FragmentManager fragmentManager;
    Fragment fragment = null;

    public UserAdapter(ArrayList<UserBarrios> arrayList, Context context ) {
        this.arrayList = arrayList;
        this.context = context;
    }

    @Override
    public UserViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_barrios, parent, false);
        UserViewHolder userViewHolder = new UserViewHolder(view, context);
        return userViewHolder;
    }


    @Override
    public void onBindViewHolder(UserViewHolder holder, int position) {
        UserBarrios user = arrayList.get(position);

        holder.textBarrio.setText(user.getBarrio());
        holder.textRuta.setText(user.getRuta());
    }

    @Override
    public int getItemCount() { return arrayList.size(); }
```

Figura 20.
Código UserAdapter

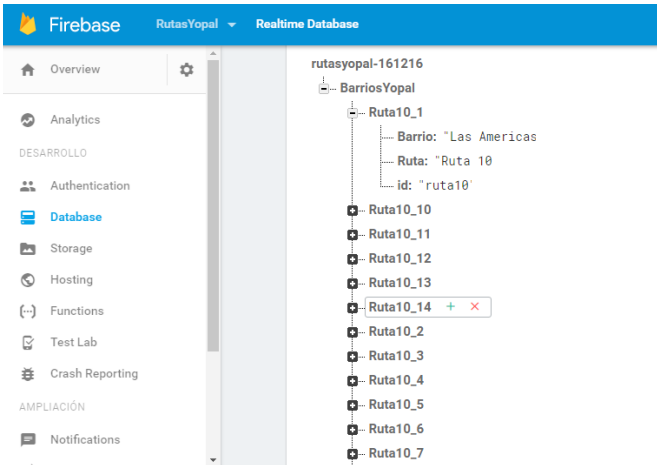
La Fig. 21 enseña parte del código que contiene las variables usadas en el recyclerview, estas deben coincidir con las que se encuentran en la base de datos para evitar conflictos.

Figura 21.
Código UserBarrios

```
public class UserBarrios {

    private String Barrio;
    private String Ruta;
    private String id;

    public UserBarrios() {

    }

    public UserBarrios(String barrio, String ruta, String id) {
        this.setBarrio(barrio);
        this.setRuta(ruta);
        this.setId(id);
    }

    public String getBarrio() { return Barrio; }

    public void setBarrio(String barrio) { Barrio = barrio; }

    public String getRuta() { return Ruta; }

    public void setRuta(String ruta) { Ruta = ruta; }

    public String getId() { return id; }

    public void setId(String id) { this.id = id; }
}
```

La Fig. 22 es el formato JSON, para la creación de la base de datos en Firebase.

Figura 22.
Base de datos Firebase



La Fig. 23 representa el ejemplo para habilitar el permiso de lectura, con el fin de que la aplicación acceda a la información de los barrios, que se encuentra almacenada en la base de datos.

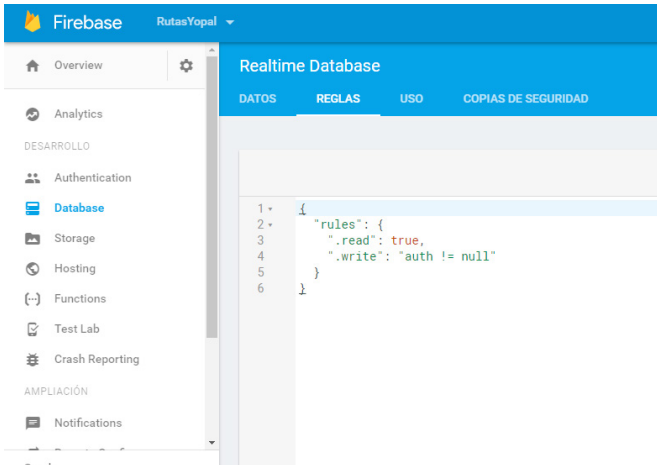


Figura 23.
Habilitar permiso de lectura

En la Fig. 24 se enseña la lista de barrios, y en la 25 se realiza el filtrado escribiendo el nombre del barrio.

Figura 24.
Interfaz de lista de barrios

Figura 25.
Interfaz de búsqueda de barrios

Fase de pruebas

Se realizan cuando se han completado las etapas anteriores y se inicia el correcto funcionamiento del sistema desarrollado. Esta etapa es crucial en el proceso porque permite detectar errores y corregirlos.

Para realizar las pruebas unitarias se utilizó AndroidTest, que viene incluido en Androd Studio . Este test se puede ejecutar en el JVM local o mediante una prueba instrumentada que se ejecute en un dispositivo

Pruebas unitarias: Las pruebas realizadas se hicieron en unas secciones del código para verificar el correcto funcionamiento de los permisos que se habilitan en Android y verificar la conexión que se tiene (wi-fi o datos). Estos permisos son el de INTERNET, ACCESS_FIN_LOCATION Y ACCES_NETWORK_STATE.

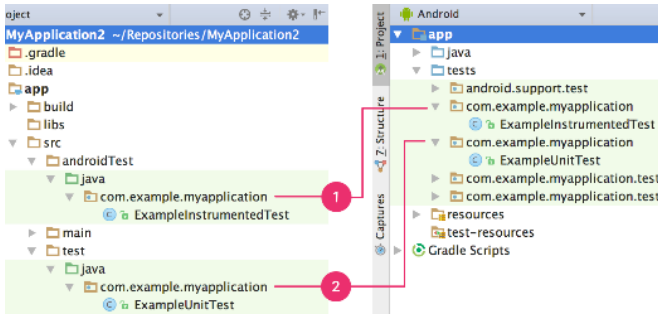


Figura 26.
Las pruebas instrumentadas (1) de tu proyecto y (2) las pruebas JVM locales [7]

En la Fig. 27 se enseña el código utilizado para llevar a cabo la prueba de conectividad, y en la Fig. 28 se presenta el resultado obtenido.

```
blic class TestConnectivity extends AndroidTestCase {

  private ConnectivityManager connectivityManager;

  @Override
  protected void setUp() throws Exception {
      super.setUp();

      connectivityManager = (ConnectivityManager) getContext().getSystemService(Context.CONNECTIV
      assertNotNull(connectivityManager);
  }

  public void testActiveConexion() {
      NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();
      assertNotNull(networkInfo);
      assertEquals(NetworkInfo.State.CONNECTED, networkInfo.getState());
  }
```

d Monitor

Figura 27.
Código prueba de conectividad

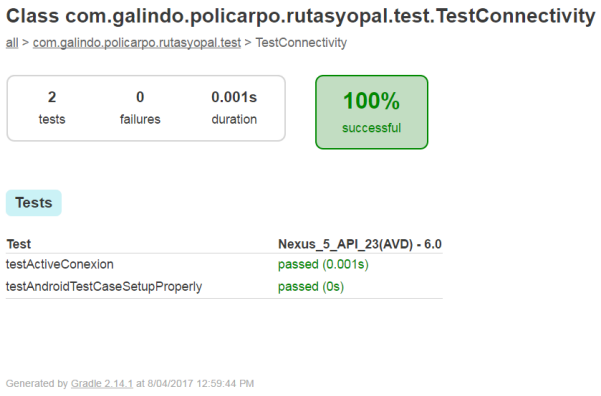


Figura 28.
Resultados prueba de conectividad

La Fig. 29 reproduce el código utilizado para llevar a cabo la prueba de los permisos que requiere la aplicación y la 30 registra un error en AccesFineLocation, debido a que no estaba habilitado el GPS en el celular.

```
ublic class testUserPermissions extends AndroidTestCase {

    String packName;
    PackageManager packageManager;

    public void setUp() throws Exception {
        super.setUp();

        packName = "com.galindo.policarpo.rutasyopal";
    }

    public void testPermissionAccesNetworState() {
        this.packageManager = this.getContext().getPackageManager();
        int res = this.packageManager.checkPermission("android.permission.ACCESS_NETWORK_STATE", packName);
        assertEquals(res, PackageManager.PERMISSION_GRANTED);
    }

    public void testPermissionInternet() {
        this.packageManager = this.getContext().getPackageManager();
        int res = this.packageManager.checkPermission("android.permission.INTERNET", packName);
        assertEquals(res, PackageManager.PERMISSION_GRANTED);
    }

    public void testAccesFineLocation() {
        this.packageManager = this.getContext().getPackageManager();
        int res = this.packageManager.checkPermission("android.permission.ACCESS_FINE_LOCATION", packName);
        assertEquals(res, PackageManager.PERMISSION_GRANTED);
    }
```

Figura 29.
Código prueba de permisos

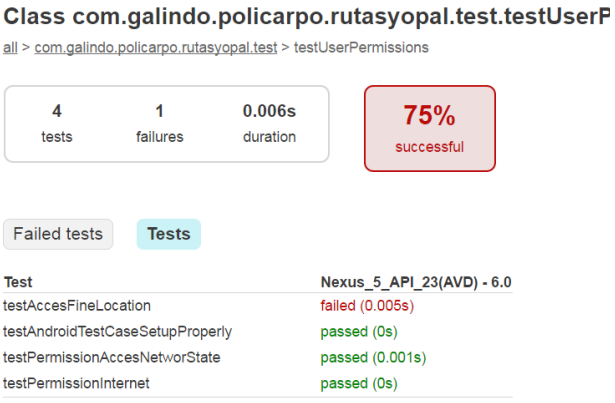


Figura 30.
Resultados prueba de permisos

En la Fig. 30 se registra la prueba de conexión entre Firebase y Android Studio, lo que generó un error que no permitía que la información de la base de datos alojada en Firebase se visualizara en la pantalla del celular. Una vez realizadas las correcciones necesarias. el sistema no genero más mensajes de error.

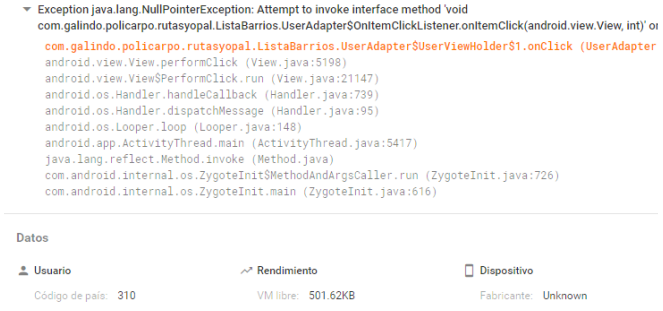


Figura 31.
Prueba UserAdapter

Pruebas de aceptación: Se realizan con el usuario para verificar el funcionamiento de la aplicación y que este dé la aprobación.

Tabla 20.
Prueba de aceptación 01

PRUEBA DE ACEPTACIÓN 01	Seleccionar ruta
Descripción	Comprobar que los usuarios puedan registrarse con su cuenta de Google, en caso de no tenerla contar con la opción añadir cuenta y finalmente acceder a la aplicación.
Especificaciones de entrada	Cuenta de Google (correo, contraseña).
Criterios de aceptación	Que el usuario acceda a la aplicación haciendo uso de su cuenta de Google.
Si cumple	

Tabla 21.
Prueba de aceptación 02

PRUEBA DE ACEPTACIÓN 02	Seleccionar ruta
Descripción	Comprobar que las rutas de las busetas se puedan visualizar en el mapa.
Especificaciones de entrada	Estar logueado en la aplicación.
Criterios de aceptación	Que el usuario pueda seleccionar la ruta que quiere ver y que esta se muestre en el mapa.
Si cumple	

PRUEBA DE ACEPTACIÓN 03	Seleccionar ruta
Descripción	Comprobar que se visualice la lista de barrios por los que pasan las busetas y realizar la búsqueda por barrio.
Especificaciones de entrada	Estar logueado en la aplicación.
Criterios de aceptación	que el usuario tenga acceso a la lista de barrios y que el resultado de la búsqueda se corresponda con el barrio que se escribe.
Si cumple	

Tabla 22.
Prueba de aceptación 03

PRUEBA DE ACEPTACIÓN 04	Seleccionar ruta
Descripción	Comprobar que el usuario pueda salir de la aplicación.
Especificaciones de entrada	Estar logueado en la aplicación.
Criterios de aceptación	Que el usuario pueda salir de su cuenta seleccionando en el menú la opción “cerrar sesión”.
Si cumple	

Tabla 23.
Prueba de aceptación 04

Conclusiones

A través de la implementación de la metodología XP, llevando a cabo cada una de sus fases (planeación, diseño, codificación y pruebas), se desarrolló la aplicación de manera óptima y ágil.

La utilización de diagramas UML (Lenguaje Unificado de Modelado) permitió comprender la interacción entre el usuario y el sistema, así como la forma en que el sistema debe responder a las peticiones del usuario, presentando de manera visual el comportamiento esperado para iniciar la fase de programación.

Android Studio es de código libre y cuenta con una comunidad que ofrece sus conocimientos para ayudar a los programadores novatos, lo que significó un importante respaldo a la realización del proyecto para solucionar dudas al momento de realizar la programación.

La API (Application Programming Interface) Google Maps permitió integrar el mapa de la ciudad de Yopal y hacer el trazado mediante coordenadas para que el trayecto del autobús fuera visible en la aplicación [6, 7, 8].

Por último, el inicio de sesión con la cuenta de Google de la aplicación facilita al usuario el proceso de registro e ingreso, lo cual conlleva a una mayor usabilidad

Referencias Bibliográficas

1. "Supervisión de la tecnología: Las aplicaciones móviles alcanzan un nuevo hito", Itu.int, 2009. [Online]. Disponible en: https://www.itu.int/net/itunews/issues/2009/06/04-es.aspx
2. K. Kendall and J. Kendall, Análisis y diseño de sistemas, 6th ed. México: Pearson, 2005, p. 20. [Online]. Disponible en: https://books.google.com.co/s?id=SXD3OWcApcIC&printsec=frontcover&dq=base+de+datos&hl=es&sa=X&sqi=2&redir_esc=y#v=onepage&q&f=false
3. " Definición de casos de uso ", Ibm.com, (2017, Mar 03). [Online]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SSWSR9_11.0.0/com.ibm.pim.dev.doc/pim_tsk_arc_definingusecases.html
4. "Diagramas de secuencia UML: Referencia", Msdn.microsoft.com, (2017, Mar 03). [Online]. Disponible en: https://msdn.microsoft.com/es-co/library/dd409377.aspx
5. "Diagramas de actividades UML: Referencia", Msdn.microsoft.com, (2017, Mar 03). [Online]. Disponible en: https://msdn.microsoft.com/es-co/library/dd409360.aspx
6. L. Ferrer Castellanos, K. González Insignares and L. Mendoza Vega, "La innovación como factor clave para mejorar la competitividad de las pymes en el departamento del Atlántico, Colombia", Dictamen Libre, no. 16, pp. 21-36, 2015.
7. P. app, "Probar tu app | Android Studio", Developer.android.com, (2017, Mar 16). [Online]. Disponible en: https://developer.android.com/studio/test/index.html?hl=es-419
8. C. tarjetas, "Crear listas y tarjetas | Android Developers", Developer.android.com, (2017, Mar 16). [Online]. Disponible en: https://developer.android.com/training/material/lists-cards.html?hl=es

Bibliografía de consulta

K. Beck, Una explicación de la programación extrema. Aceptar el cambio. Madrid: Pearson Educación, S.A., 2002.

C. Studio, "Conoce Android Studio | Android Studio", Developer.android.com, (2017, Feb 22). [Online]. Disponible en: https://developer.android.com/studio/intro/index.html?hl=es-419

About - Google Maps, Google.com, (2016, Feb 20). [Online]. Disponible en: https://www.google.com/maps/about/

Diseños | Android Developers", Developer.android.com, (2017, Abr 25. [Online]. Disponible en: https://developer.android.com/guide/topics/ui/declaring-layout.html?hl=es-419

Firebase, Firebase, (2016, Abr 16). [Online]. Disponible en: https://firebase.google.com/?hl=es-419

GOOGLE PLAY STORE, Google, (2016, Mar 02). [Online]. Disponible en internet: https://play.google.com/about/developer-content-policy.html

Este artículo se cita

P. Galindo., y M. Suárez, "Diseño e Implementación de una Aplicación Móvil Android para el Seguimiento de Rutas de Transporte Urbano en el Municipio de Yopal", Investigación e Innovación en Ingenierias, vol. 5, n°. 2, pp. 138-173, 2017