



Instituto Politécnico Nacional
Escuela Superior de Cómputo



❖ **INTEGRANTES DEL EQUIPO:**

- Barrera Perez Carlos Tonatihu
 - López Higuera Antonio
 - López Sánchez Andrés
 - Monroy Martos Elioth

❖ **UNIDAD DE APRENDIZAJE:** Application Development for Mobile Devices

❖ **PROFESOR:** Alfredo Sigfrido Cifuentes Alvarez.

❖ **PROYECTO SEGUNDO PARCIAL:** Aplicación Android-Web, blog con tutoriales sobre Android/Java.

❖ **FECHA:** 15 de abril de 2019.

❖ **Versión del reporte:** 1.0

❖ **GRUPO:** 3CM8.

Índice

Índice	2
Objetivo	3
Objetivos específicos	3
Conceptos	3
Desarrollo	4
Descripción	4
Software y hardware utilizados	4
Código fuente	5
Pruebas	13
Conclusiones	21
Dificultades encontradas	21
Posibles aplicaciones	21
Conclusiones individuales	22
Barrera Perez Carlos Tonatihu	22
López Higuera Antonio	22
López Sánchez Andrés	22
Monroy Martos Elioth	22
Bibliografía	23

Objetivo

Desarrollar una aplicación web y móvil, cliente-servidor, tomando como base un tema específico de programación, agregando seguridad (login), inserción, búsqueda, modificación y eliminación en una base de datos. Tanto de forma local (usando SQLite) tanto como de forma remota.

Objetivos específicos

- Diseño de una aplicación con la arquitectura cliente-servidor.
- Proteger la página web por medio de un login.
- Proporcionar un contenido educativo de algún tema de programación en Android.
- Implementación de una base de datos para gestionar las cuentas de usuario.
- Permitir la inserción de comentarios por usuarios registrados.
- Permitir modificación de los datos del usuario.
- Permitir la eliminación de cuentas de usuario.
- Almacenar información localmente en la aplicación Android.
- Almacenar información de forma remota usando un servidor web.

Conceptos

Para el entendimiento de esta proyecto es necesario conocer los siguientes conceptos los cuales fueron clave durante el desarrollo de esta aplicación.

- **Java Server Pages (JSP).** Es una tecnología web que permite la generación de contenido dinámico. Está conformado por etiquetas HTML y etiquetas JSP (expresiones, declaraciones y scriptlets). [3]
- **Servlet.** Es una tecnología web basada en la generación de contenido dinámico, ésta puede atender peticiones y dar respuesta a las mismas. [3]
- **Java.** Es un lenguaje de programación orientado a objetos, compilado e interpretado por su máquina virtual, Java puede ser utilizado en prácticamente cualquier tipo de proyecto debido a su portabilidad.[1]
- **Android.** Es un sistema operativo para dispositivos móviles, basado en el núcleo de linux, diseñado principalmente para dispositivos táctiles.[2]
- **Activity Android.** Una actividades es un punto de entrada para la interacción con el usuario. Representa una pantalla con una interfaz de usuario.[2]
- **Android Manifest.** Es un archivo que describe la información esencial de la aplicación a las herramientas de construcción de Android, al sistema operativo y a Google Play. En esta se describe el nombre del paquete, los componentes de la aplicación (incluye todas las actividades, servicios,

content provides, entre otros), los permisos que requiere la aplicación, entre otras declaraciones esenciales. [2]

- **Aplicación Web.** Es un sistema de software con una capa de negocio, y su *front end* es en gran parte es mostrado por un sistema web.[3]

Desarrollo

Descripción

El proyecto consiste en realizar una aplicación móvil la cual permita a los usuarios participar en un blog, el cual habla sobre tutoriales de programación Java y Android. Las acciones que puede realizar un usuario (altas, bajas, cambios, consultar) quedan almacenadas en una base de datos local en el teléfono Android, a la vez que cada una de estas acciones se ven reflejadas en un servidor web. El cual funciona como una base de datos remota sobre la cual se ven reflejados los cambios.

Software y hardware utilizados

A continuación, se presenta una lista del software y hardware usado para la elaboración del proyecto:

- **Android Studio 3.4.** Android IDE para el desarrollo de aplicaciones para el sistema operativo Android.
- **NetBeans.** Es un IDE que permite el desarrollo de aplicaciones Java, permitiendo la integración con Apache Tomcat para el desarrollo de aplicaciones web.
- **Apache Tomcat.** Es un servidor web y contenedor de servlets, provee una implementación de JSP y Servlet API, además de todas las API's para JavaEE y algunas de Java SE..
- **Teléfono con Android 5+.** Usado para realizar la pruebas correspondientes.
- **Java.** Lenguaje de programación multipropósito con el cual fue programada la aplicación móvil y el servidor web.

Código fuente

A continuación, se presenta el código más importante elaborado, para el desarrollo del proyecto.

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.androidclient">
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.READ_PROFILE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".view.ComentarioFormActivity"
            android:noHistory="true"></activity>
        <activity android:name=".view.ComentarioActivity" />
        <activity android:name=".view.PostActivity" />
        <activity
            android:name=".view.UsuarioFormActivity"
            android:noHistory="true" />
        <activity
            android:name=".view.PostFormActivity"
            android:noHistory="true" />
        <activity
            android:name=".view.HomeActivity"
            android:label="@string/title_activity_home"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity android:name=".view.WebViewActivity" />
        <activity android:name=".view.LoginActivity" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

HomeActivity.java:

```
package com.example.androidclient.view;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.support.design.widget.TabLayout;
import android.support.design.widget.FloatingActionButton;
```

```

import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.ViewPager;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import com.example.androidclient.R;
import com.example.androidclient.dao.PostDAO;
import com.example.androidclient.dto.Post;
import com.example.androidclient.dto.Usuario;
import com.example.androidclient.utilerias.AdapterPost;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
public class HomeActivity extends AppCompatActivity {
    private Usuario usuario;
    private PostDAO dao;
    private List<Post> listaPost;
    private PlaceholderFragment pageHome;
    private PlaceholderFragment pageJava;
    private PlaceholderFragment pageAndroid;
    private SectionsPagerAdapter mSectionsPagerAdapter;
    /**
     * The {@link ViewPager} that will host the section contents.
     */
    private ViewPager mViewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        //GetPost Values
        dao = new PostDAO(this);
        usuario = (Usuario) getIntent().getSerializableExtra("usuario");
        listaPost = dao.readAll();
        pageHome = new PlaceholderFragment( listaPost, usuario);
        pageJava = new PlaceholderFragment(
            filterCategoria(listaPost, "Java"), usuario );
        pageAndroid = new PlaceholderFragment(
            filterCategoria(listaPost, "Android"), usuario );

        //
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        // Create the adapter that will return a fragment for each of the
three
        // primary sections of the activity.
        mSectionsPagerAdapter = new
SectionsPagerAdapter(getSupportFragmentManager());

```

```

        // Set up the ViewPager with the sections adapter.
        mViewPager = (ViewPager) findViewById(R.id.container);
        mViewPager.setAdapter(mSectionsPagerAdapter);

        TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);

        mViewPager.addOnPageChangeListener(new
        TabLayout.TabLayoutOnPageChangeListener(tabLayout));
        tabLayout.addOnTabSelectedListener(new
        TabLayout.ViewPagerOnTabSelectedListener(mViewPager));

        FloatingActionButton fab = (FloatingActionButton)
        findViewById(R.id.fab);
        fab.setOnClickListener(setListener( PostFormActivity.class ));
    }

    @Override
    public void onRestart() {
        super.onRestart();
        listaPost = dao.readAll();
        List<Post> post1 = new ArrayList<>();
        List<Post> post2 = new ArrayList<>();
        post1.addAll(listaPost);
        post2.addAll(listaPost);

        pageHome.updateAdapter( listaPost);
        pageJava.updateAdapter(
            filterCategoria( post1 , "Java"));
        pageAndroid.updateAdapter(
            filterCategoria( post2, "Android"));
    }

    @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_home, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            Intent itn = new Intent(HomeActivity.this,
            UsuarioFormActivity.class);
            itn.putExtra("usuario", usuario);
            startActivity(itn);
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    private View.OnClickListener setListener(final Class<?> clase){
        return new View.OnClickListener() {

```

```

        public void onClick(View v) {
            Intent itn = new Intent(HomeActivity.this, clase);
            itn.putExtra("usuario", usuario);
            startActivity(itn);
        }
    };
}

private List<Post> filterCategoria( List<Post> lista, String categoria
){
    List<Post> nuevaLista = new ArrayList<>();
    for (Post post: lista){
        if(post.getCategoriaPost().equals(categoria))
            nuevaLista.add( post );
    }
    return nuevaLista;
}

/**
 * A placeholder fragment containing a simple view.
 */
@SuppressWarnings("ValidFragment")
public static class PlaceholderFragment extends Fragment {

    public RecyclerView recyclerPost;
    public List<Post> listaPost = new ArrayList<>();
    private Usuario usuarioInternFragment;

    /**
     * The fragment argument representing the section number for this
     * fragment.
     */
    private static final String ARG_SECTION_NUMBER = "section_number";

    public PlaceholderFragment(List<Post> listaPost, Usuario
usuarioInternFragment) {
        this.listaPost = listaPost;
        this.usuarioInternFragment = usuarioInternFragment;
    }

    @Override
    public View onCreateView( LayoutInflater inflater, ViewGroup
container,
                                Bundle savedInstanceState ) {
        View rootView = inflater.inflate( R.layout.fragment_home,
container, false );
        recyclerPost = (RecyclerView) rootView.findViewById(
R.id.xrecyclerId );
        recyclerPost.setLayoutManager( new
LinearLayoutManager(this.getContext()) );
        Context c = this.getContext();
        AdapterPost adapterPost = new AdapterPost(listaPost);
        adapterPost.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent itn = new Intent(v.getContext(),
PostActivity.class);

```



```

        Post postAux =
listaPost.get(recyclerPost.getChildAdapterPosition(v));
        itn.putExtra("post", postAux);
        itn.putExtra("usuario", usuarioInternFragment);
        //itn.putExtra("usuario", usuario);
        startActivity(itn);
    }
});
recyclerPost.setAdapter(adapterPost);
return rootView;
}

public void updateAdapter(List<Post> listaPostAux) {
    AdapterPost adapterPost = new AdapterPost(listaPostAux);
    this.listaPost = listaPostAux;
    adapterPost.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent itn = new Intent(v.getContext(),
PostActivity.class);
            Post postAux =
listaPost.get(recyclerPost.getChildAdapterPosition(v));
            itn.putExtra("post", postAux);
            itn.putExtra("usuario", usuarioInternFragment);
            //itn.putExtra("usuario", usuario);
            startActivity(itn);
        }
    });
    System.out.println("===== ");
    System.out.println("listaPost : " + listaPostAux);
    recyclerPost.setAdapter(adapterPost);
}

}

/**
 * A {@link FragmentPagerAdapter} that returns a fragment
corresponding to
 * one of the sections/tabs/pages.
 */
public class SectionsPagerAdapter extends FragmentPagerAdapter {

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        switch (position) {
            case 0:
                return pageHome;
            case 1:
                return pageJava;
            case 2:
                return pageAndroid;
            default:
                return pageHome;
        }
    }
}

```

```

    }

    private List<Post> filterCategoria( List<Post> lista, String
categoria ) {
        List<Post> nuevaLista = new ArrayList<>();
        for (Post post: lista) {
            if (post.getCategoriaPost().equals(categoria))
                nuevaLista.add(post);
        }
        return nuevaLista;
    }
    @Override
    public int getCount() {
        return 3;
    }
    @Override
    public CharSequence getPageTitle(int position) {
        String section = null;
        switch (position) {
            case 0:
                section = "Home";
                break;
            case 1:
                section = "Java";
                break;
            case 2:
                section = "Android";
                break;
        }
        return section;
    }
}
}

```

activity_home.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main_content"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:context=".view.HomeActivity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/appbar_padding_top"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"

```

```

        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:layout_weight="1"
        android:background="?attr/colorPrimary"
        app:layout_scrollFlags="scroll|enterAlways"
        app:popupTheme="@style/AppTheme.PopupOverlay"
        app:title="@string/app_name">

</android.support.v7.widget.Toolbar>

<android.support.design.widget.TabLayout
    android:id="@+id/tabs"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:tabTextColor="#000">

    <android.support.design.widget.TabItem
        android:id="@+id/tabItem"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tab_text_home" />

    <android.support.design.widget.TabItem
        android:id="@+id/tabItem2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tab_text_java" />

    <android.support.design.widget.TabItem
        android:id="@+id/tabItem3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tab_text_android" />

</android.support.design.widget.TabLayout>
</android.support.design.widget.AppBarLayout>

<android.support.v4.view.ViewPager
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior" />

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end|bottom"
    android:backgroundTint="@color/colorSuccess"
    android:layout_margin="@dimen/fab_margin"
    app:srcCompat="@android:drawable/ic_input_add" />

</android.support.design.widget.CoordinatorLayout>

```

WebViewActivity.java:

```

package com.example.androidclient.view;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebSettings;
import android.webkit.WebView;
import com.example.androidclient.R;
import com.example.androidclient.utilerias.WebViewClientImpl;
public class WebViewActivity extends AppCompatActivity {
    private WebView webView = null;
    String urlServidor;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_web_view);
        this.webView = (WebView) findViewById(R.id.webview);
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        urlServidor = getIntent().getExtras().getString("urlServidor");
        WebViewClientImpl webViewClient = new WebViewClientImpl(this,
urlServidor);
        webView.setWebViewClient(webViewClient);
        webView.loadUrl(urlServidor);
    }
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if ((keyCode == KeyEvent.KEYCODE_BACK) && this.webView.canGoBack())
        {
            this.webView.goBack();
            return true;
        }
        return super.onKeyDown(keyCode, event);
    }
}

```

activity_web_view.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".view.WebViewActivity">
    <WebView
        android:id="@+id/webview"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</android.support.constraint.ConstraintLayout>

```

Pruebas

A continuación se muestran algunas capturas de pantalla de la aplicación móvil.

Android Client

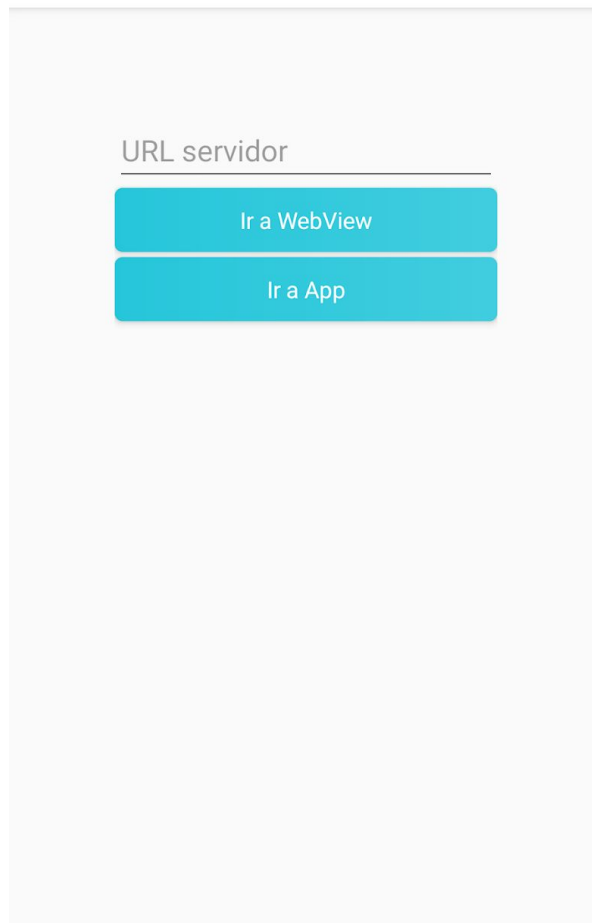
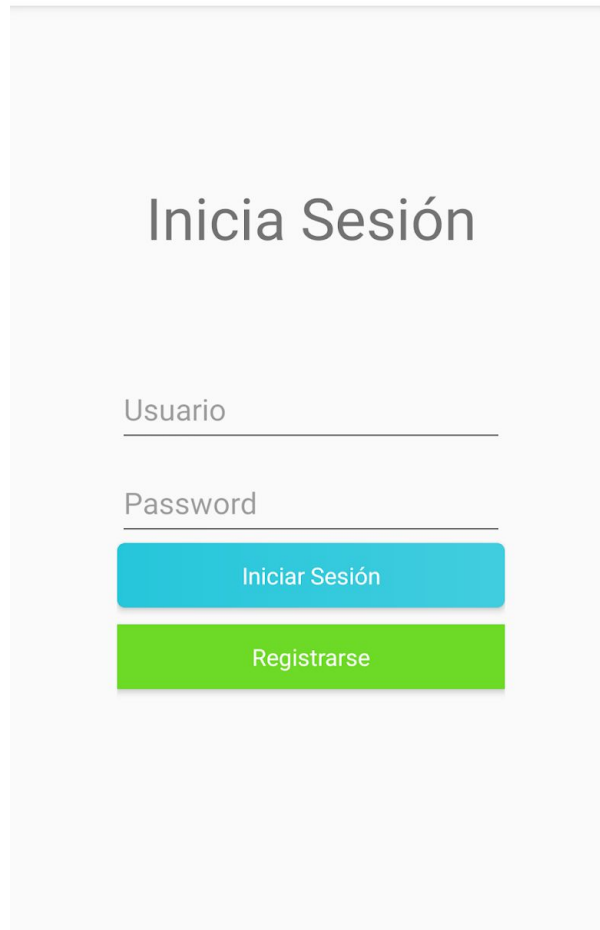


Figura 1. Elección inicial de aplicación.



The image shows a login screen for an Android client. At the top, the title "Inicia Sesión" is displayed in a large, dark gray font. Below the title, there are two input fields: "Usuario" and "Password", each with a horizontal line underneath. Below the "Password" field, there are two buttons: a blue button labeled "Iniciar Sesión" and a green button labeled "Registrarse". The entire form is centered on a light gray background.

Figura 2. Inicio de sesión.

Android Client

Información Usuario

Nombre
Elioth

Apellido
Monroy

Email
eliothmonroy@yahoo.com.mx

Password
1234

Agregar

Figura 3. Registro de usuario.



Figura 4. Pantalla de inicio o home.

Android Client

Información Post

Titulo

Java

▼

Resumen

Contenido

Agregar

Figura 5. Creación de un post.

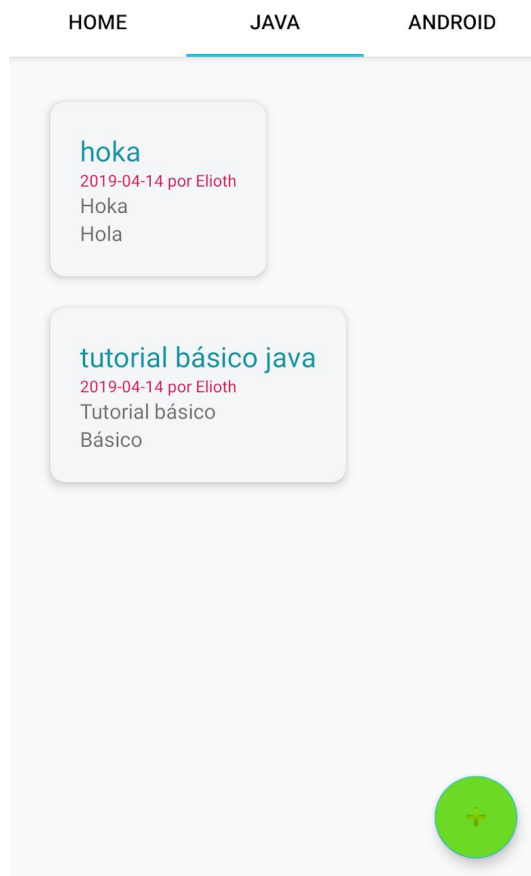


Figura 6. Pestaña de Java.

Android Client

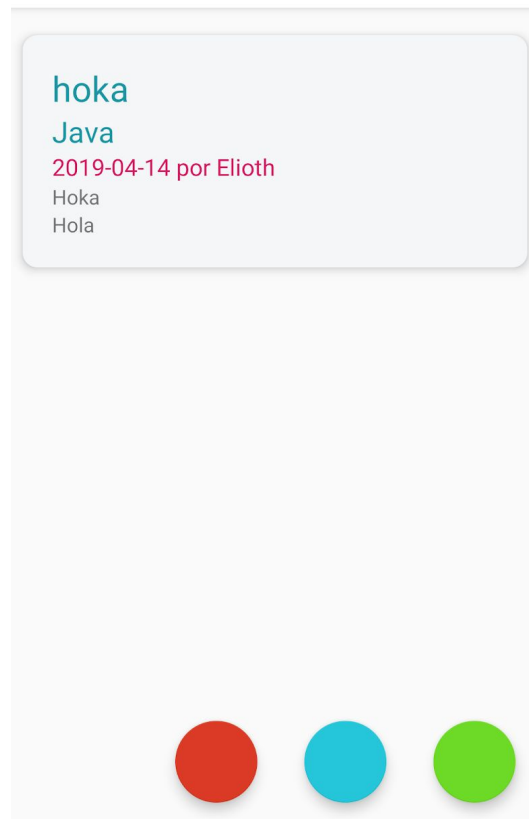


Figura 7. Post seleccionado.

Android Client

Información Post

Título

tutorial básico java

Java

▼

Resumen

Tutorial básico

Contenido

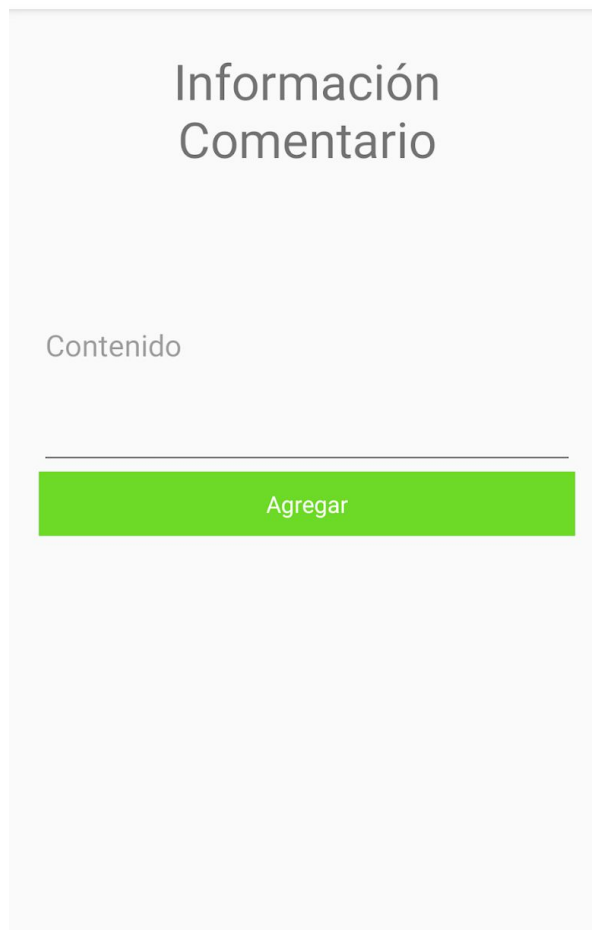
Básico

Eliminar

Actualizar

Figura 8. Editar post.

Android Client



The screenshot shows a mobile application interface with a light gray background. At the top, the text "Información Comentario" is displayed in a large, dark gray font. Below this, the word "Contenido" appears in a smaller, lighter gray font. A horizontal line separates the text area from a bright green rectangular button at the bottom, which contains the white text "Agregar".

Figura 9. Agregar comentario.

Conclusiones

Dificultades encontradas

Al principio, resultó complicado comprender las diferentes partes que componen el proyecto, ya que una vez que fue elaborado el servidor web y el sistema web. la forma en que pensamos había que implementarlo en Android era mediante el uso de webviews, posteriormente, entendimos que había que hacer una aplicación Android la cual se conectará al servidor de forma independiente con el uso de webviews.

Posibles aplicaciones

En la actualidad, la tendencia es que las aplicaciones móviles no solo funcionen de manera local en un dispositivo, si no que los usuarios esperan que las mismas se

conecten con otros servicios de Internet, ya sea para realizar un respaldo de información (como sucede con Whatsapp por ejemplo), o para tener una sincronización de la información entre distintos dispositivos, como puede ser el caso de aplicaciones como Facebook o Instagram, las cuales tienen la capacidad de almacenar información de forma local y al encontrarse con conexión a Internet subir la misma a sus servidores.

Conclusiones individuales

Barrera Perez Carlos Tonatihu

Este proyecto es una pequeña muestra del cómo pueden funcionar las aplicaciones móviles hoy en día, ya que a veces no basta con que en el teléfono se tenga toda la información que se está procesando, debido a que se puede llegar a perder, es por esto que hacer que trabaje junto a un servidor o una aplicación web en general le agrega un nivel mayor de integridad a los datos.

Y no solo esto, muchos servicios que utilizamos en la actualidad nos proporcionan tanto la versión web como la móvil.

López Higuera Antonio

El realizar una buena aplicación web, tanto *front end* como *back end* permite poder transportarla a cualquier otra plataforma, en este caso el desarrollar la aplicación de manera que se adaptara a la pantalla permite poder visualizarla en el celular sin tener que desarrollar una aplicación específicamente para el móvil.

López Sánchez Andrés

La realización del proyecto resultó un conjunto de todos los conocimientos visto al rededor del curso pues tuvimos que hacer uso de cada uno de los elementos de android que hicimos en tareas. Adicionalmente existió una dificultad al crear una arquitectura que pudiera conectar una aplicación móvil y realizar operaciones dentro de la ésta y a la vez actualizar en un servidor independiente, pues tuvimos que investigar un poco el cómo dar permisos en Android y ver otros detalles pequeños. Como conclusión puedo decir que ya tengo las bases para desarrollar una aplicación que pueda ser utilizada en un ámbito personal, o inclusive para una pequeña empresa.

Monroy Martos Elioth

Con este proyecto aplicamos varios de los conocimientos que adquirimos durante el curso al momento de desarrollar la aplicación móvil, así como también de otras materias al desarrollar el servidor web. Pudimos ver un poco la aplicación que se le

pueden dar a la aplicaciones móviles usando servicios “en la nube”, ya que nuestro servidor actuaba como una base de datos remota sobre la cual se hacía una copia de toda la información local que manejamos internamente en la aplicación móvil. Por lo cual, podemos decir que de cierta forma, es como si hiciéramos un respaldo de esa información, simulando un servicio en la nube como alguno de los actuales como Google Drive y demás.

Bibliografía

- [1] M. Lozano Ortega and B. Bonev, "Introducción a Android", *Curso de Android y Java para Dispositivos Móviles*, 2010. [Online]. Disponible en: <http://www.jtech.ua.es/apuntes/ajdm2010/sesiones/sesion09-apuntes.html>. [Consultado: 10- Abril- 2019].
- [2] “Application Fundamentals”, Android Developer, 2019. [Online]. Disponible en: <https://developer.android.com/guide/components/fundamentals>. [Consultado: 10-Abril-2019].
- [3] “Learn JSP”, Tutorialspoint, 2019. [Online]. Disponible en <https://www.javatpoint.com/jsp-tutorial>. [Consultado: 10-Abril-2019].

Manual de usuario

A continuación se muestran las diversas pantallas que componen a la aplicación móvil, así como una explicación de las diversas funcionalidades que se pueden usar en cada una.

Android Client

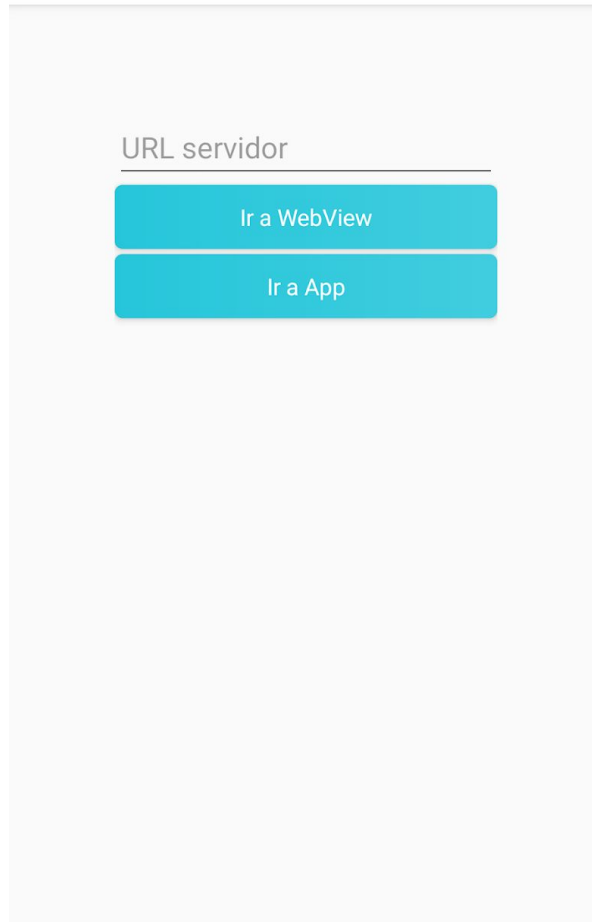


Figura 1. Elección inicial de aplicación.

En la Figura 1, se muestra la pantalla “Inicio”, en la cual se puede ingresar la url de un servidor web y conectarse al mismo mediante un webview, al presionar el botón “Ir a WebView”. De igual manera, al presionar el botón “Ir a App” se muestra la versión Android de la aplicación web, con la cual se puede interactuar con la información almacenada localmente. Al presionar ese botón, se muestra la pantalla “Iniciar sesión”.

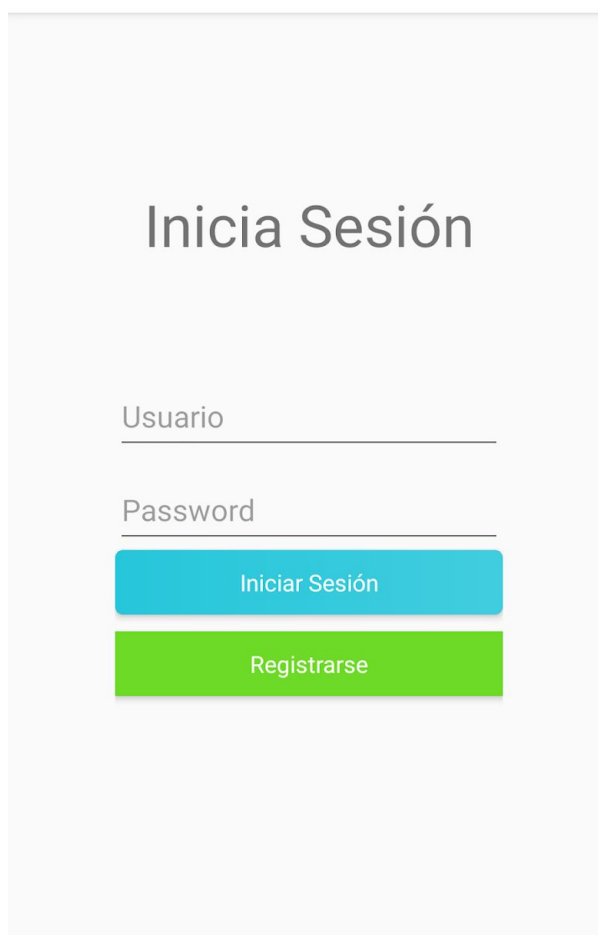


Figura 2. Inicio de sesión.

En la Figura 2, se muestra la pantalla “Iniciar sesión”, se accede a esta pantalla al presionar el botón “Ir a App” de la pantalla “Inicio”. En esta pantalla se permite al usuario acceder al sistema al usar su nombre y contraseña. Para esto, debe de haberse registrado al sistema. En esta misma pantalla, se muestran dos botones, una para iniciar sesión, y otro para registrarse en el sistema. Al llenar la información solicitada y presionar el botón “Iniciar Sesión”, se valida en el sistema que exista el usuario y en caso de ser correcta la información se muestra la pantalla “Home”. Al presionar el botón “Registrarse” se muestra la pantalla “Registro”.

Android Client

Información Usuario

Nombre
Elioth

Apellido
Monroy

Email
eliothmonroy@yahoo.com.mx

Password
1234

Agregar

Figura 3. Registro de usuario.

En la Figura 3, se muestra la pantalla “Registro”, en la cual se muestran los campos necesarios para que un usuario se registre en el sistema, esta información es almacenada al presionar el botón “Agregar”. Una vez que un usuario se registra en el sistema puede iniciar sesión.

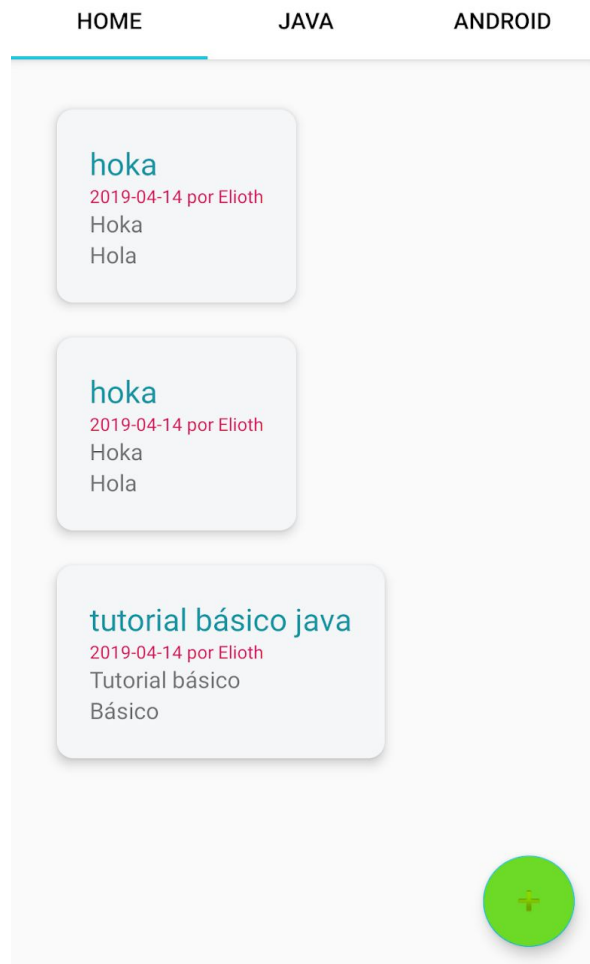


Figura 4. Pantalla de inicio o home.

En la Figura 5, se muestra la pantalla “Home”, esta puede ser accesada al presionar el botón “Iniciar sesión” en la pantalla “Iniciar sesión”. En esta pantalla se muestran las publicaciones que existen en el sistema, así como también las diferentes pestañas en las que pueden ser clasificadas las publicaciones. A su vez cada publicación puede ser accesada al presionar sobre alguna de ellas, y también pueden ser creadas otras publicaciones al presionar sobre el botón “Crear Post” que se encuentra en la esquina inferior derecha.

Android Client

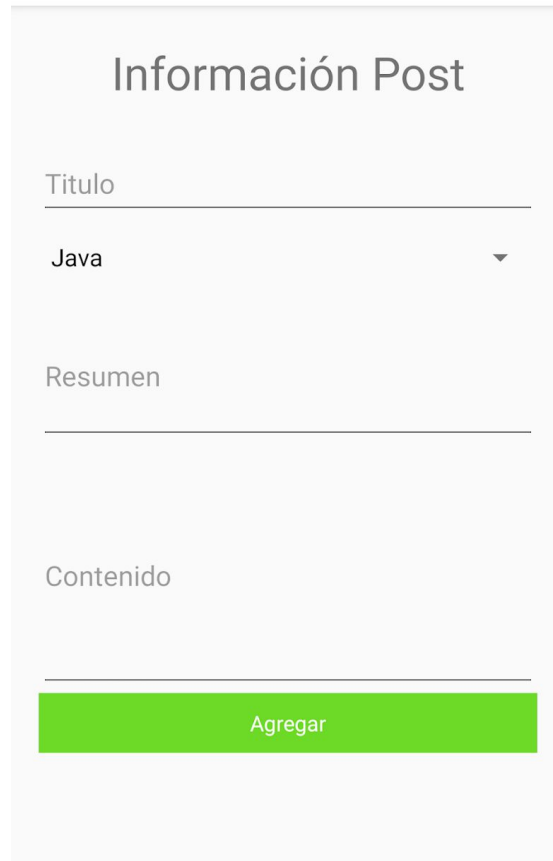


Figura 5. Creación de un post.

En la Figura 5, se muestra la pantalla “Crear Post”, a la cual se puede acceder al presionar el botón “Crear Post” de la pantalla “Home”. En esta pantalla se muestran los campos que son necesarios para crear un post, al finalizar se puede presionar el botón “Agregar” con el cual se puede guardar la publicación.

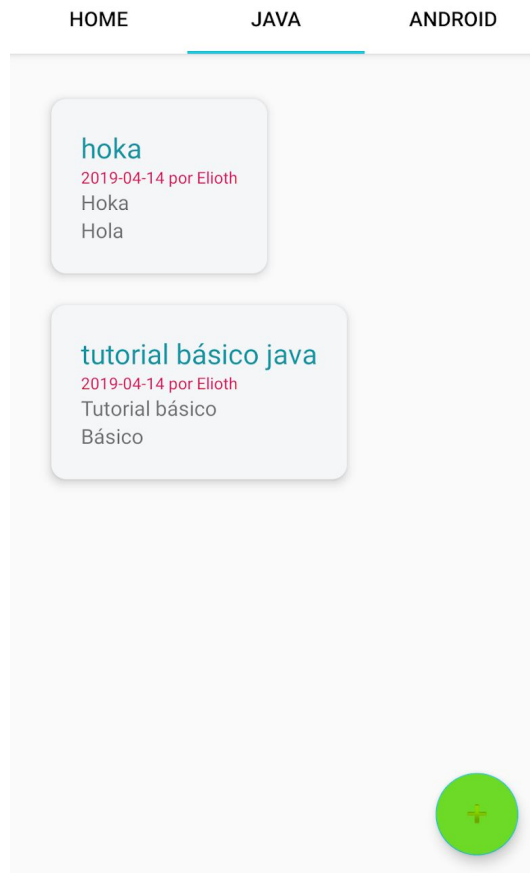


Figura 6. Pestaña de Java.

En la Figura 6, se muestra la pantalla “Pestaña”, la cual es como la pantalla “Home” pero con la diferencia de que se muestran solamente las publicaciones correspondientes a la pestaña en la que el usuario se encuentra. Esto quiere decir que solo se mostrarán las publicaciones que cumplan con la etiqueta de la pestaña. De igual forma, se puede presionar sobre alguna de las publicaciones para visualizar su información y poder crear un comentario.

Android Client

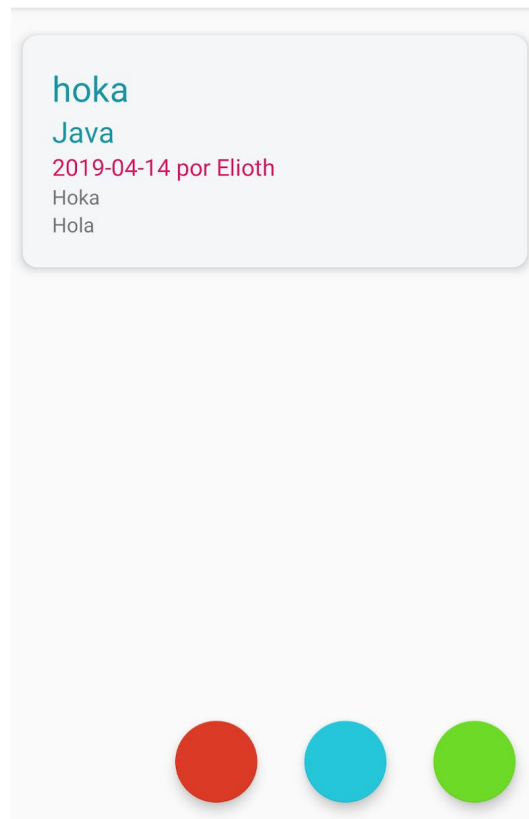


Figura 7. Post seleccionado.

En la Figura 7, se muestra la pantalla “Visualizar Post”, la cual muestra la información detallada de la publicación seleccionada, además de que muestra tres botones con acciones posibles sobre cada una de las publicaciones, el botón rojo “Eliminar post” permite al creador del post eliminar el mismo. El botón azul “Editar post” permite al autor del post cambiar la información del post seleccionado, para esto se muestra la pantalla “Editar post”. Finalmente, el botón verde “Agregar comentario”, permite a un usuario añadir un comentario sobre el post. Esto se puede realizar en la pantalla “Agregar comentario”.

Android Client

Información Post

Título

tutorial básico java

Java

▼

Resumen

Tutorial básico

Contenido

Básico

Eliminar

Actualizar

Figura 8. Editar post.

En la Figura 8, se muestra la pantalla “Editar Post”, desde la cual, un autor puede editar la información de un post que haya creado, además se muestran dos botones los cuales le permiten eliminar y actualizar el post respectivamente.

Android Client

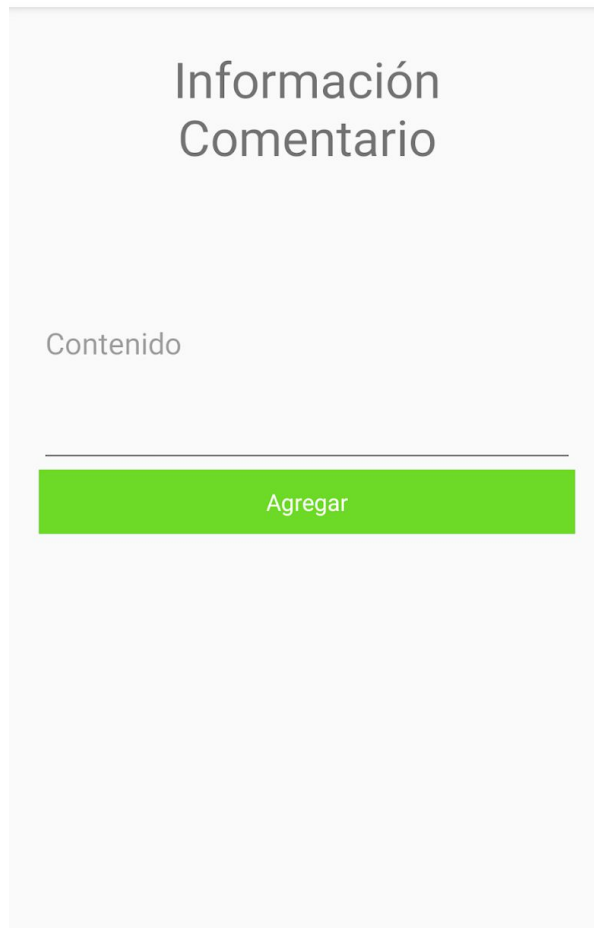


Figura 9. Agregar comentario.

En la Figura 9, se muestra la pantalla “Agregar comentario”, al llenar el campo “Contenido” se puede presionar el botón “Agregar”, el cual almacena de forma local y remota el comentario creado por el usuario sobre el post seleccionado. Para realizar esto, el usuario debe estar loggeado en el sistema.