
Especificación de requisitos de software

Proyecto: Workflix Empresarial

1er Cuatrimestre TSDWAD - ISPC

2024

Ficha del documento

Fecha	Revisión	Autor	Referencias de cambios.
2024/04/26	Alvarez, Natalia	ALVAREZ, Natalia Beatriz DELGADO Marcelo DIAZ, Yandira GIMÉNEZ, Diego Fabián GÓMEZ, Mariela Beatriz LÓPEZ, Erick LORENTI, Valentino TRONCOSO, Braian	Se comienza con la construcción de la documentación. Se redactan historias de usuarios. Se agregan requerimientos funcionales y no funcionales. Se completa cuadro de Sprint 0

Contenido

<u>FICHA DEL DOCUMENTO</u>	2
<u>CONTENIDO</u>	3
<u>1 INTRODUCCIÓN</u>	4
<u>1.1 Propósito</u>	4
<u>1.2 Alcance</u>	4
<u>1.3 Limitaciones</u>	4
<u>1.4 Personal involucrado</u>	4
<u>1.5 Definiciones, acrónimos y abreviaturas</u>	6
<u>1.6 Referencias</u>	6
<u>1.7 Resumen</u>	6
<u>2 DESCRIPCIÓN GENERAL</u>	6
<u>2.1 Perspectiva del producto</u>	6
<u>2.2 Características de los usuarios</u>	7
<u>2.3 Restricciones</u>	7
<u>2.4 Suposiciones y Dependencias</u>	7
<u>3 REQUISITOS ESPECÍFICOS</u>	9
<u>3.1 Product Backlog</u>	9
<u>3.2 Requerimientos Funcionales</u>	9
<u>3.3 Requerimientos No Funcionales</u>	11
<u>3.4 Sprints</u>	14
<u>3.4.1 Sprint 0</u>	14
<u>3.4.2 Sprint 1</u>	14
<u>3.4.3 Sprint 2</u>	14
<u>4 Anexo</u>	15

1 Introducción

Este documento es una Especificación de Requisitos de Software (ERS) para el Sistema de información para la gestión de procesos y control de la aplicación **Workflix**. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, para el desarrollo de un sistema de información web (WorkflixWeb) que permitirá que trabajadores de mantenimiento y servicios (plomeros, pintores, albañiles, etc) puedan ser contactados dentro de la plataforma y ser calificados por los clientes una vez terminados los trabajos, de esta manera se espera aumentar su visibilidad y transparentar su nivel de profesionalismo para adquirir más y mejores trabajos dentro de la plataforma a la hora de la búsqueda.

También estableceremos las especificaciones funcionales de la aplicación móvil (WorkflixApp) que permitirá que trabajadores de mantenimiento y servicios puedan darse de alta, modificar su perfil y agregar servicios. A su vez podrán ser contactados, por fuera de la plataforma, por los visitantes de la misma, mediante un enlace.

1.2 Alcance

Esta especificación está dirigida al usuario principal del sistema y contendrá los requerimientos analizados y acordados con el usuario como así también información técnica sobre el proceso de desarrollo en sí.

La aplicación funcionará como mediador entre quien ofrece un servicio doméstico y quien lo necesita.

El sistema permitirá al usuario Profesional darse de alta en la aplicación móvil, así como iniciar sesión, modificar datos personales y profesionales.

El sistema permitirá al usuario Cliente, darse de alta en el sitio web, así como iniciar sesión, consultar catálogo y contratar Profesionales.

El sistema permitirá, además, al usuario Administrador dar de baja al usuario si este no respetara las normas de ética y buenas costumbres.

El solo hecho de utilizar el sistema debería ser suficiente para darle credibilidad al sitio.

1.3 Limitaciones

- Solo podrán darse de alta o iniciar sesión en la aplicación móvil los usuarios Profesionales.
- El usuario visitante solo se podrá contactar con el usuario Profesional por fuera de la aplicación móvil.
- La Aplicación Móvil no soportará pasarela de pago.
- El sistema no se responsabilizará por operaciones realizadas a partir del contacto entre usuarios.

1.4 Personal involucrado

Nombre	Alvarez, Natalia Beatriz
Rol	Scrum Master / Desarrolladora
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	natalia.humanista@gmail.com

Nombre	Delgado, Marcelo
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	marceloedelgado@gmail.com

Nombre	Diaz, Yandira
Rol	Desarrolladora
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	yandiradiaz1@gmail.com

Nombre	Giménez, Diego
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	fabian.diego@gmail.com

Nombre	Gómez, Mariela Beatriz
Rol	Desarrolladora
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	marielagomez_95@hotmail.com

Nombre	López, Erick
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	erick.kever@gmail.com

Nombre	Lorenti, Valentino
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	lorentivalentino@gmail.com

Nombre	Troncoso, Braian
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Diseño y programación del sistema
Información de contacto	troncosobraianaxel@gmail.com

1.5 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
SIS-I	Sistema de Información Web para la Gestión de Procesos Administrativos y Académicos
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
FTP	Protocolo de Transferencia de Archivos
US	Historias de usuarios
TK	Tareas realizadas
Administrador	Persona que utilizará el sistema para gestionar bajas de profesionales
Visitante	Persona que utilizará el sistemas para ver catálogo de profesionales
Profesional	Persona que utilizará el sistema para ofrecer sus servicios
Cliente	Persona que utilizará la página web para contratar a profesionales

1.6 Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.7 Resumen

Este documento consta de 3 (tres) secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la sección 2 se da una descripción general del sistema, con el fin de conocer la perspectiva del producto, las características de los usuarios, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen los requisitos específicos que debe satisfacer el sistema, detalle de historias de usuarios (US) y de los diferentes Sprint.

Workflix Web Institucional, funcionará como una página de aterrizaje para que los usuarios obtengan información sobre las aplicaciones (web y mobile) y sean redirigidos a ellas.

WorkflixApp es una aplicación para que los usuarios Profesionales ofrezcan sus servicios de mantenimiento domiciliario y los usuarios visitantes puedan elegir, de acuerdo a la información brindada, entre una amplia gama de profesionales.

WorkflixWeb tiene como objetivo principal facilitar la contratación de servicios por parte de los usuarios clientes y brindar la funcionalidad de suscripción a los profesionales que deseen ofrecer sus servicios en la plataforma.

2 Descripción general

2.1 Perspectiva del producto

WorkflixApp es una aplicación nativa de Android. Será un producto diseñado para ser utilizado en teléfonos inteligentes y tablets, lo que permitirá su utilización de forma rápida y eficaz. Es una aplicación que permitirá que trabajadores de mantenimiento y servicios (plomeros, pintores, albañiles, etc) puedan darse de alta, modificar su perfil y agregar servicios. Los usuarios visitantes podrán tener información de los Profesionales que se ofrecen y podrán contactarse con ellos por fuera de la aplicación.

WorkflixWeb es un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma rápida y eficaz. Está pensado como un e-commerce de servicios que proporcionará una interfaz de usuario intuitiva y fácil de usar, permitiendo a los usuarios clientes buscar servicios, visualizar detalles, realizar pagos seguros y recibir confirmaciones de compra. Además, los profesionales podrán crear perfiles, listar sus servicios, establecer precios y recibir pagos por sus servicios.

2.2 Características de los usuarios

Tipo de usuario	Administrador
Formación	Manejo de herramientas informáticas
Actividades	Control y manejo del sistema en general, alta, baja y modificación de Profesionales

Tipo de usuario	Visitante
Formación	Manejo de aplicaciones móviles
Actividades	Observar servicios y profesionales disponibles. Ponerse en contacto con usuarios proveedores de servicios.

Tipo de usuario	Profesional
Formación	Manejo de aplicaciones móviles
Actividades	Darse de alta e iniciar sesión. Modificación de datos personales y de servicios que brinda.

Tipo de usuario	Cliente
Formación	Manejo de entornos web
Actividades	Darse de alta e iniciar sesión. Modificar datos personales. Contratar y puntuar profesionales

2.3 Restricciones

- Interfaz para ser usada con internet (solamente).
- Lenguajes y tecnologías en uso: Android, Java, SpringBoot, MySQL, HTML, CSS, Bootstrap, JavaScript, Django, Angular y Python
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño e implementación sencillos.
- No se contemplan operaciones paralelas por fuera de las descriptas en el sistema.

2.4 Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma

3 Requisitos específicos

3.1 Product Backlog

#US01: Como usuario cliente, en la página web, me gustaría que en la página inicial se muestre la opción para descargar la aplicación en el celular.

#US02: Como usuario cliente, me gustaría poder visualizar en la web y en la app los mismos profesionales para elegir cuál contratar.

#US03 Como usuario cliente, me gustaría que la web y la app muestren el mismo diseño para que sea consistente.

#US04 Como usuario cliente, en la app mobile, me gustaría tener un botón que me redirija a la página web institucional para obtener más información.

#US05 Como usuario visitante me gustaría poder visualizar los profesionales mejor calificados de la app mobile para elegir el profesional adecuado.

#US06 Como usuario registrado me gustaría poder recuperar mi contraseña de ingreso en el caso de olvidarla para poder ingresar nuevamente al sitio o a la aplicación móvil

#US07 Como usuario visitante, en la aplicación móvil, me gustaría poder filtrar los profesionales para poder elegir con más facilidad según el rubro que busco.

#US08 Como usuario visitante, en la aplicación móvil, me gustaría poder contactar directamente por whatsapp a los profesionales, para que sea más sencillo contactarlos.

3.2 Requerimientos funcionales

Identificación del requerimiento	RF01
Nombre del Requerimiento	Catálogo de profesionales
Características	Ver opciones de profesionales que ofrecen servicios
Descripción del requerimiento	Plantilla con los profesionales que ofrecen sus servicios
Requerimiento no funcional	RNF01 - RNF02 - RNF03 - RNF04
Prioridad	Media

Identificación del requerimiento	RF02
Nombre del Requerimiento	Registrar usuarios
Características	Los usuarios Profesionales deberán registrarse para poder ofrecer sus servicios
Descripción del requerimiento	El sistema permitirá al usuario Administrador registrarse. Mediante formulario de registración el

	usuario debe suministrar datos como: Nombre, Apellido, E-mail, Usuario, Password, etc
Requerimiento no funcional	RNF01 - RNF02 - RNF03 - RNF04 - RNF05 - RNF07
Prioridad	Alta

Identificación del requerimiento	RF03
Nombre del Requerimiento	Autenticar usuarios
Características	Los usuarios deberán identificarse para acceder al sistema
Descripción del requerimiento	El sistema podrá ser consultado por cualquier usuario dependiendo su nivel de accesibilidad.
Requerimiento no funcional	RNF01 - RNF02 - RNF04 - RNF05 - RNF06 - RNF07
Prioridad	Alta

Identificación del requerimiento	RF04
Nombre del Requerimiento	Perfil
Características	Los usuarios podrán acceder a su perfil
Descripción del requerimiento	El sistema permitirá que los usuarios accedan a su perfil para ver, agregar o modificar información.
Requerimiento no funcional	RNF01 - RNF02 - RNF05 - RNF07
Prioridad	Media/Alta

Identificación del requerimiento	RF05
Nombre del Requerimiento	Recuperar contraseña
Características	El usuario podrá solicitar una nueva contraseña en caso de olvidarla
Descripción del requerimiento	Se brindará un formulario de recuperación de contraseña en caso de necesitarlo.
Requerimiento no funcional	RNF01 - RNF02 - RNF03 - RNF05 - RNF07
Prioridad	Media

Identificación del requerimiento	RF06
Nombre del Requerimiento	Filtro
Características	El sistema permitirá al usuario buscar por categoría profesional
Descripción del requerimiento	Se podrá buscar profesional por rama (plomero, albañil, gasista, etc)
Requerimiento no funcional	RNF01 - RNF02
Prioridad	Baja

Identificación del requerimiento	RF07
Nombre del Requerimiento	Calificación
Características	El sistema permitirá al usuario calificar a un profesional
Descripción del requerimiento	Como forma de recomendación a otros usuarios
Requerimiento no funcional	RNF01
Prioridad	Baja

Identificación del requerimiento	RF08
Nombre del Requerimiento	Detalle Profesional
Características	Ampliar la información de los profesionales.
Descripción del requerimiento	Para poder visualizar la información completa del profesional
Requerimiento no funcional	RNF01 - RNF02 - RNF03 - RNF04 - RNF07
Prioridad	Media

Identificación del requerimiento	RF09
Nombre del Requerimiento	Gestión de usuarios
Características	El sistema permitirá al Administrador modificar datos del profesional
Descripción del requerimiento	Permite al administrador modificar datos de los profesionales o darlos de baja.
Requerimiento no funcional	RNF01 - RNF03 - RNF04 - RNF05 - RNF06 - RNF07
Prioridad	Media/Alta

3.3 Requerimientos no funcionales

Identificación del requerimiento	RNF01
Nombre del Requerimiento	Interfaz del sistema.
Características	El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento	El sistema debe tener una interfaz de uso intuitiva y sencilla
Prioridad	Alta

Identificación del requerimiento	RNF02
Nombre del Requerimiento	Diseño de la interfaz a la característica de la web.

Proyecto Integrador
Especificación de requisitos de software

Características	El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web Workflix
Descripción del requerimiento	La interfaz de usuario debe ajustarse a las características de la web Workflix
Prioridad	Alta

Identificación del requerimiento	RNF03
Nombre del Requerimiento	Mantenimiento
Características	El sistema debe utilizar prácticas adecuadas y estandarizada
Descripción del requerimiento	Facilitar actualizaciones y correcciones de errores
Prioridad	Media

Identificación del requerimiento	RNF04
Nombre del Requerimiento	Disponibilidad
Características	El sistema tendrá que estar en funcionamiento las 24 horas los 7 días de la semana. Ya que es una aplicación diseñada para la carga de datos y comunicación entre usuarios
Descripción del requerimiento	La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado.
Prioridad	Alta

Identificación del requerimiento	RNF05
Nombre del Requerimiento	Seguridad
Características	El sistema garantizara a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
Descripción del requerimiento	Garantizar la seguridad del sistema con respecto a la información y datos que se manejan.
Prioridad	Alta

Identificación del requerimiento	RNF06
Nombre del Requerimiento	Nivel de Usuario
Características	Garantizara al usuario el acceso de información de acuerdo al nivel que posee.
Descripción del requerimiento	Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas

Proyecto Integrador
Especificación de requisitos de software

Prioridad	Alta
------------------	------

Identificación del requerimiento	RNF07
Nombre del Requerimiento	Fiabilidad
Características	El sistema debe tener un manejo adecuado de errores y caídas
Descripción del requerimiento	El sistema debe tener un tiempo promedio entre fallas de al menos 1000 horas.
Prioridad	Media

3.4 Sprints.

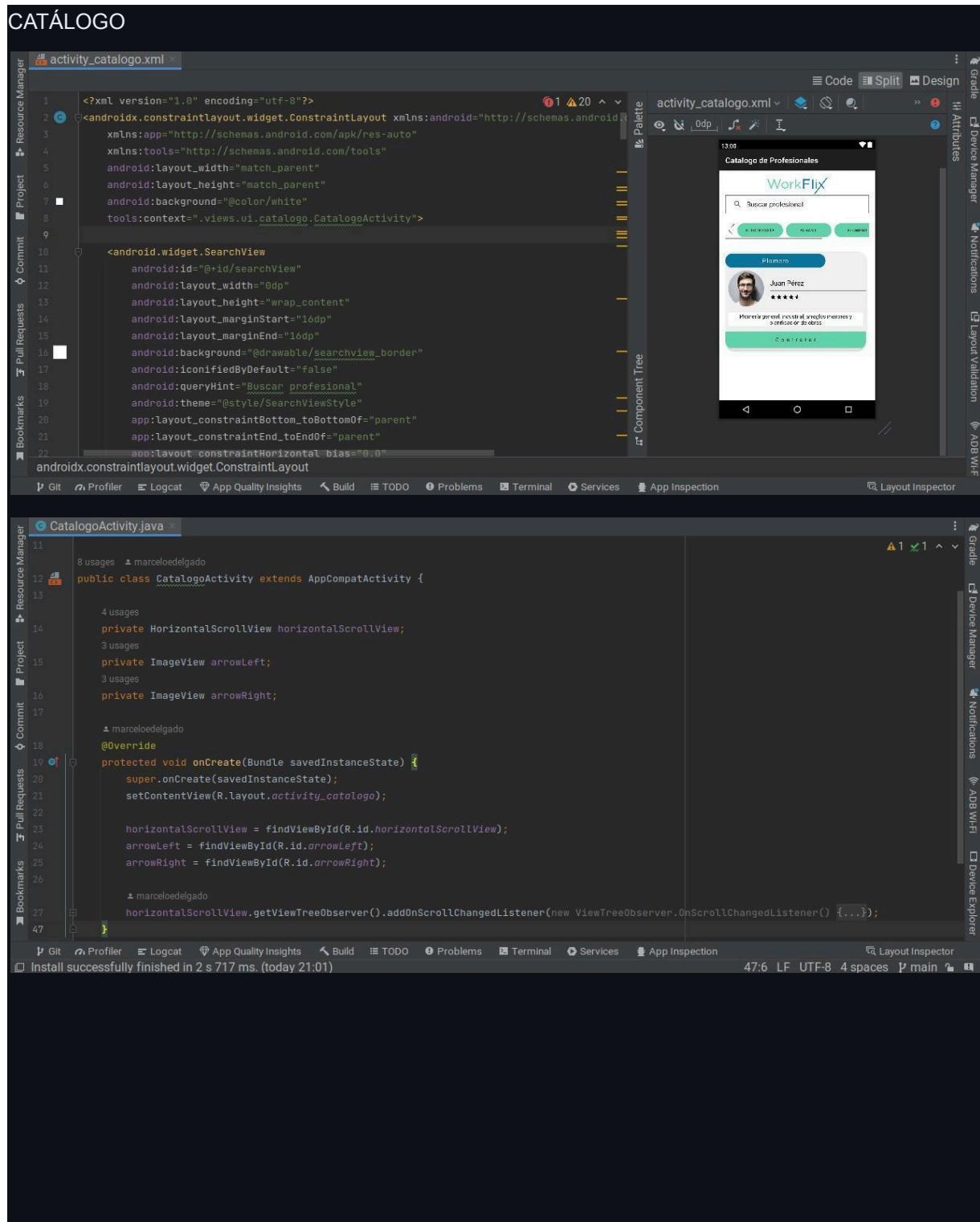
N° de sprint	0
Sprint Backlog	#US01 - #US02 - #US03 - #US04 - #US05 - #US06 - #US07 - #US08
Responsabilidades	Definir los roles de los miembros del equipo Scrum (Scrum master y Developer team). Documentar en la Wiki del Proyecto - Evaluar el contenido y distribución previa de la tienda para que se adecue al nuevo desarrollo de e-commerce / carrito de compras, analizando los requerimientos necesarios. Lo mismo ocurrirá con la App - Plantear Historias de Usuarios en base a los requerimientos del proyecto teniendo en cuenta una redacción y nomenclatura adecuada, ej “#US01 Como usuario quiero registrarme en el sitio web para comprar uno o varios productos” y su respectivos criterios de aceptación. Pueden utilizar la siguiente plantilla: Plantilla Historias de Usuario - Definir la documentación del proyecto mediante el documento IEEE830 - Crear un proyecto estilo kanban en Github - Definir la estructura de páginas en la Wiki del repositorio en github a fin de poder documentar: Nombre y apellido de los integrantes del equipo como así también los roles de cada quién, registro de ceremonias de scrum: planning, review y retrospective (para esta última es importante publicar además el plan de mejora a ejecutar en la siguiente iteración), documento IEEE830 del proyecto.
Calendario	CIERRE: 26/004/2024
Inconvenientes:	

N° de sprint	1
Sprint Backlog	Desarrollo de mejoras sustanciales y superadoras de la aplicación web (Front, Back y API Rest con CRUD). Tener en cuenta las rúbricas.
Responsabilidades	
Calendario	CIERRE: 13/05/2024
Inconvenientes:	

N° de sprint	2
Sprint Backlog	
Responsabilidades	Desarrollo de mejoras sustanciales y superadoras de la aplicación móvil (cinco o mas Activities con navegabilidad aplicando un CRUD). Tener en cuenta las rúbricas.
Calendario	CIERRE: 03/06/2024
Inconvenientes:	

4 Anexo

Imágenes de WorkflixApp



DASHBOARD

The image shows the Android Studio IDE with two main panels. The top panel displays the XML layout for 'dashboard_admin.xml'. The XML code defines a vertical LinearLayout with a background color of #FFFFFF and a weightSum of 10. Inside this layout is a RelativeLayout with a width of match_parent, a height of 60dp, and a weight of 2. The RelativeLayout contains six icons arranged in a 3x2 grid: Home, Profile, Search, Notifications, Recent Files, and Logout. The bottom panel shows the Java code for 'DashboardAdminActivity.java'. The code imports the necessary packages and defines the DashboardAdminActivity class, which extends AppCompatActivity. It includes an onCreate method that calls super.onCreate and setContentView, and an irInicio method that starts the MainActivity.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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=".views.ui.dashboard_admin.DashboardAdminActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#FFFFFF"
        android:orientation="vertical"
        android:weightSum="10"
        tools:context=".views.ui.dashboard_admin.DashboardAdminActivity">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:layout_weight="2"
            android:visibility="visible">

                <!-- Home icon -->
                <!-- Profile icon -->
                <!-- Search icon -->
                <!-- Notifications icon -->
                <!-- Recent Files icon -->
                <!-- Logout icon -->

            </RelativeLayout>

        </LinearLayout>

    </androidx.constraintlayout.widget.ConstraintLayout>
```

```
package tec.ispc.workflox.views.ui.dashboard_admin;

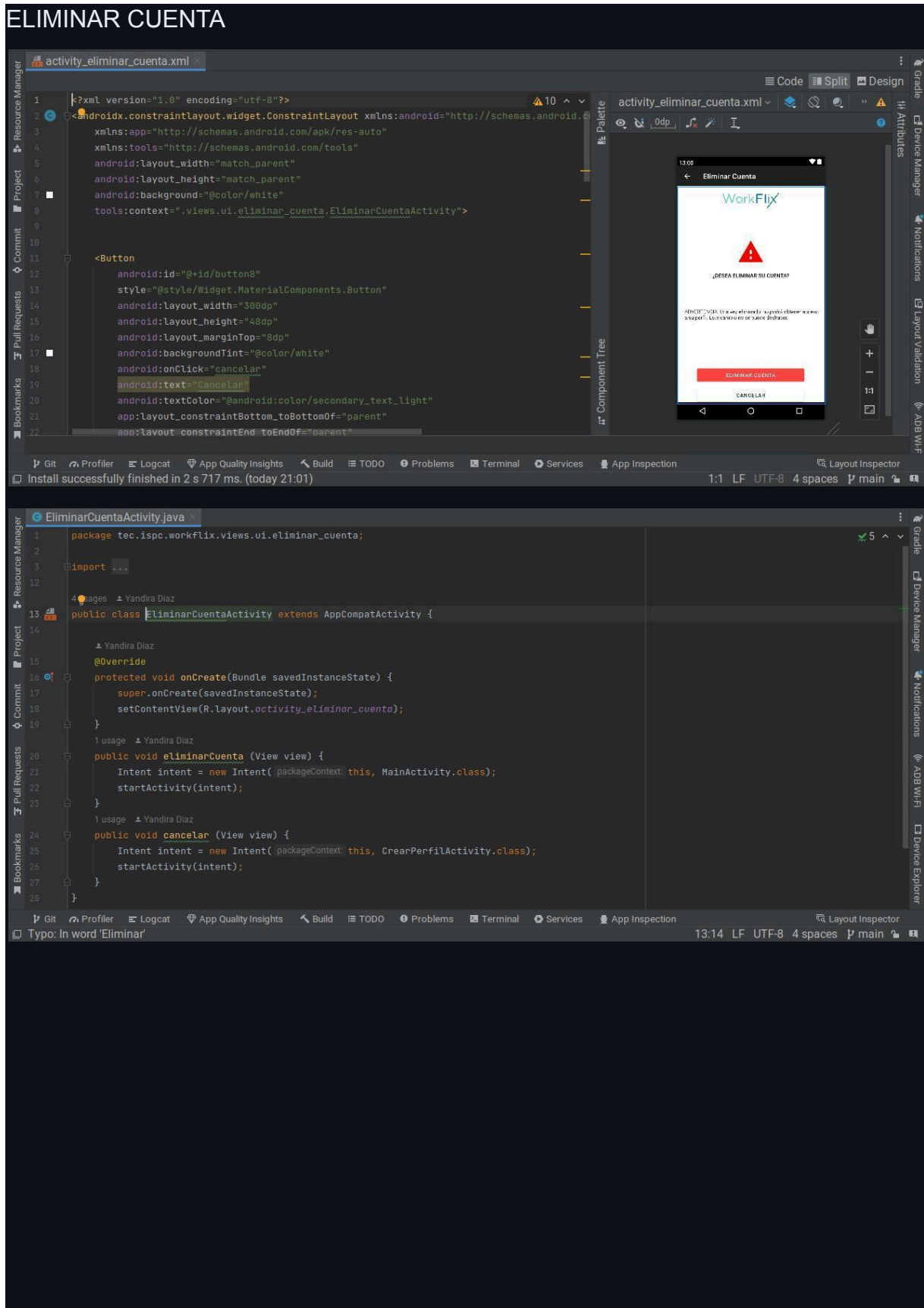
import androidx.appcompat.app.AppCompatActivity;

public class DashboardAdminActivity extends AppCompatActivity {

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

    public void irInicio(View view) {
        Intent inicioIntent = new Intent( packageContext: this, MainActivity.class);
        startActivity(inicioIntent);
    }
}
```


ELIMINAR CUENTA



The screenshot displays the Android Studio IDE with two tabs open: `activity_eliminar_cuenta.xml` and `EliminarCuentaActivity.java`.

activity_eliminar_cuenta.xml (XML Layout):

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:background="@color/white"
7      tools:context=".views.ui.eliminar_cuenta.EliminarCuentaActivity">
8
9
10
11  <Button
12      android:id="@+id/button8"
13      style="@style/Widget.MaterialComponents.Button"
14      android:layout_width="300dp"
15      android:layout_height="48dp"
16      android:layout_marginTop="8dp"
17      android:backgroundTint="@color/white"
18      android:onClick="cancelar"
19      android:text="Cancelar"
20      android:textColor="@android:color/secondary_text_light"
21      app:layout_constraintBottom_toBottomOf="parent"
22      app:layout_constraintEnd_toEndOf="parent"/>

```

EliminarCuentaActivity.java (Java Code):

```

1  package tec.ispc.workflix.views.ui.eliminar_cuenta;
2
3  import ...
4
5  public class EliminarCuentaActivity extends AppCompatActivity {
6
7      @Override
8      protected void onCreate(Bundle savedInstanceState) {
9          super.onCreate(savedInstanceState);
10         setContentView(R.layout.activity_eliminar_cuenta);
11     }
12
13     public void eliminarCuenta (View view) {
14         Intent intent = new Intent( packageContext.this, MainActivity.class);
15         startActivity(intent);
16     }
17
18     public void cancelar (View view) {
19         Intent intent = new Intent( packageContext.this, CrearPerfilActivity.class);
20         startActivity(intent);
21     }
22 }

```

The right-hand pane shows a preview of the activity, displaying a mobile app interface with a red warning triangle and the text "¿DESEA ELIMINAR SU CUENTA?". Below the text are two buttons: "ELIMINAR CUENTA" (red) and "CANCELAR" (white).

The bottom status bar indicates the app is running on a device with the following settings: 1:1 LF UTF-8 4 spaces P main.

LOGIN

The screenshot displays the Android Studio IDE with two files open: `activity_login.xml` and `LoginActivity.java`.

activity_login.xml (XML Layout):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingBottom="16dp"
    tools:context=".views.ui.login.LoginActivity">

    <Button
        android:id="@+id/inRegistro"
        style="@style/Widget.MaterialComponents.Button"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_gravity="start"
        android:layout_marginBottom="48dp"
```

LoginActivity.java (Java Code):

```
package tec.ispc.workflix.views.ui.login;

import ...

public class LoginActivity extends AppCompatActivity {

    private LoginViewModel loginViewModel;
    private ActivityLoginBinding binding;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

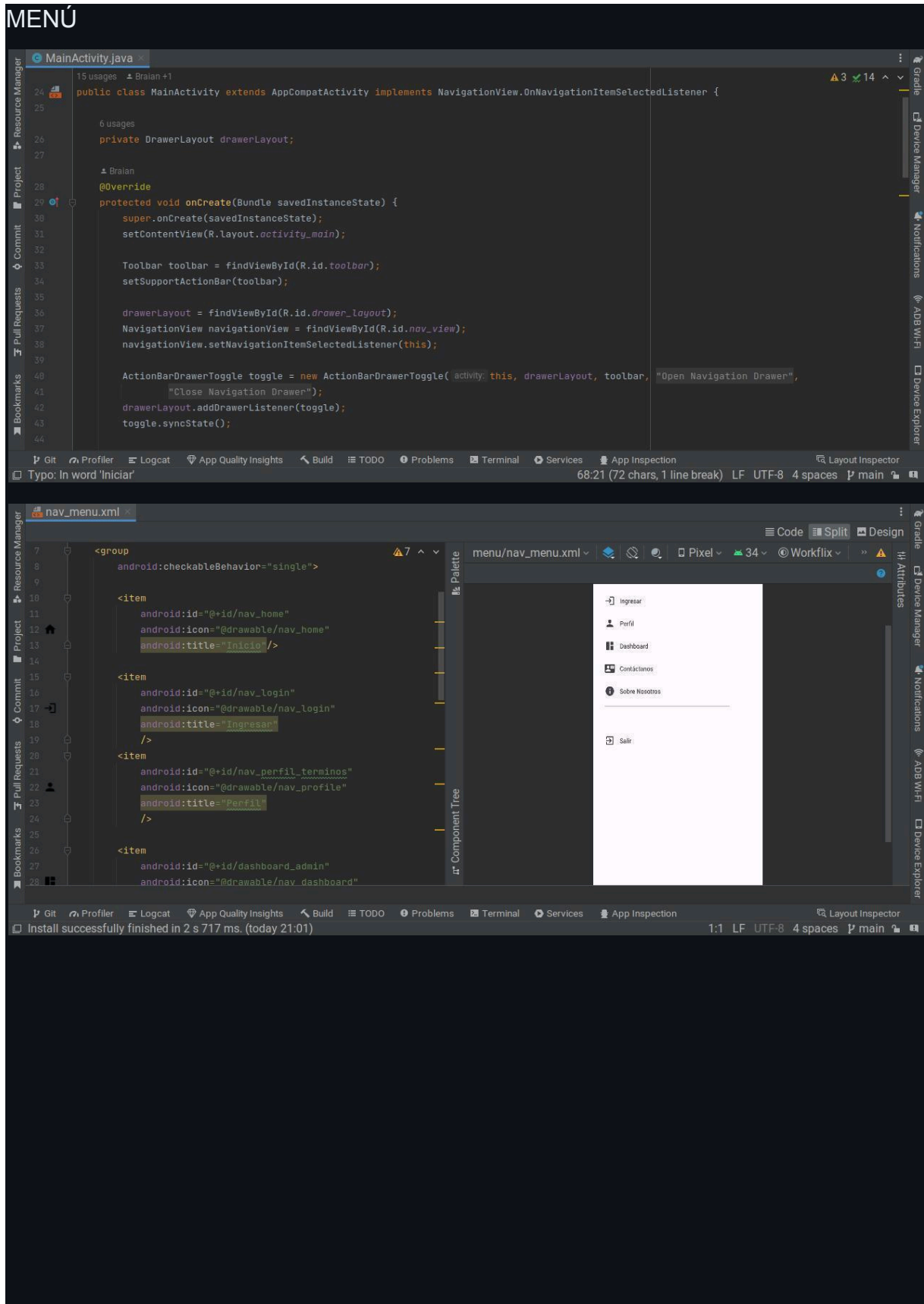
        binding = ActivityLoginBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        loginViewModel = new ViewModelProvider(owner, this, new LoginViewModelFactory())
            .get(LoginViewModel.class);

        final EditText usernameEditText = binding.username;
```

The right-hand pane shows a preview of the login screen. It features the "WorkFlix" logo, a "Login" title, and several login options: "Iniciar sesión", "Iniciar sesión", "Iniciar sesión", "Iniciar sesión", "Iniciar sesión", "Iniciar sesión", "Iniciar sesión", "Iniciar sesión", "Iniciar sesión", "Iniciar sesión".

MENÚ



The screenshot displays the Android Studio IDE with two files open: MainActivity.java and nav_menu.xml.

MainActivity.java (Lines 24-44):

```

24 public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {
25
26     private DrawerLayout drawerLayout;
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.activity_main);
32
33         Toolbar toolbar = findViewById(R.id.toolbar);
34         setSupportActionBar(toolbar);
35
36         drawerLayout = findViewById(R.id.drawer_layout);
37         NavigationView navigationView = findViewById(R.id.nav_view);
38         navigationView.setNavigationItemSelectedListener(this);
39
40         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle( this, drawerLayout, toolbar, "Open Navigation Drawer",
41             "Close Navigation Drawer");
42         drawerLayout.addDrawerListener(toggle);
43         toggle.syncState();
44     }

```

nav_menu.xml (Lines 7-28):

```

7 <group
8     android:checkableBehavior="single">
9
10     <item
11         android:id="@+id/nav_home"
12         android:icon="@drawable/nav_home"
13         android:title="Inicio"/>
14
15     <item
16         android:id="@+id/nav_login"
17         android:icon="@drawable/nav_login"
18         android:title="Ingresar"/>
19
20     <item
21         android:id="@+id/nav_perfil_terminos"
22         android:icon="@drawable/nav_profile"
23         android:title="Perfil"/>
24
25     <item
26         android:id="@+id/dashboard_admin"
27         android:icon="@drawable/nav_dashboard"
28     </item>

```

The right sidebar shows a preview of the menu layout, displaying a list of items: Ingresar, Perfil, Dashboard, Contactanos, Sobre Nosotros, and Salir.

PERFIL

The screenshot displays the Android Studio IDE with two main panels open: the XML layout editor for `activity_crear_perfil.xml` and the Java code editor for `CrearPerfilActivity.java`.

XML Layout Editor (`activity_crear_perfil.xml`):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:onClick="cerrarSesion"
    tools:context=".views.ui.perfil.CrearPerfilActivity">
    <ImageView
        android:id="@+id/imageView7"
        android:layout_width="158dp"
        android:layout_height="48dp"
        android:layout_marginTop="28dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/logo_workflix_1" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

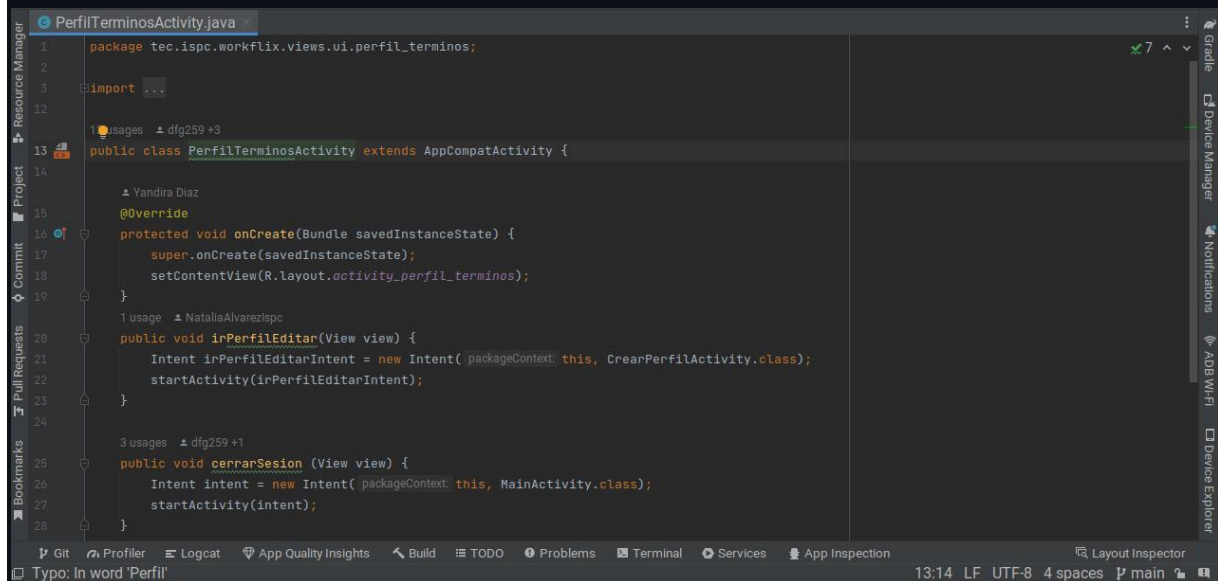
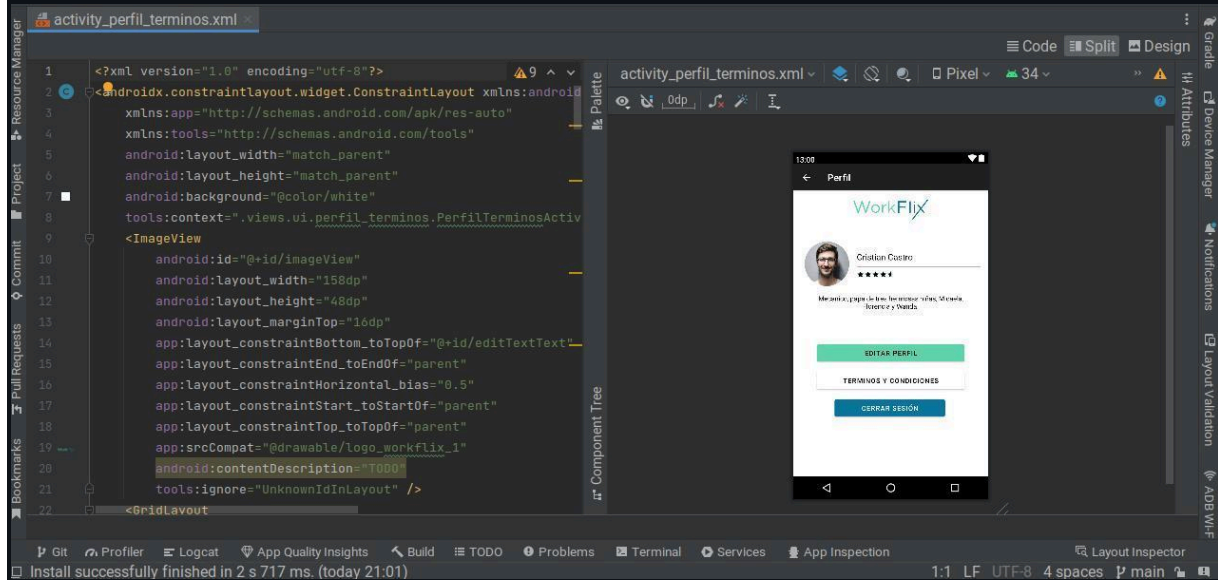
Java Code Editor (`CrearPerfilActivity.java`):

```
import ...
8 usages Mariela Gómez +2
public class CrearPerfilActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_crear_perfil);
    }
    3 usages Erick
    public void cerrarSesion (View view) {
        Intent intent = new Intent( packageContext, this, MainActivity.class);
        startActivity(intent);
    }
    1 usage Yandra Diaz
    public void cerrarCuenta (View view) {
        Intent intent = new Intent( packageContext, this, EliminarCuentaActivity.class);
        startActivity(intent);
    }
}
```

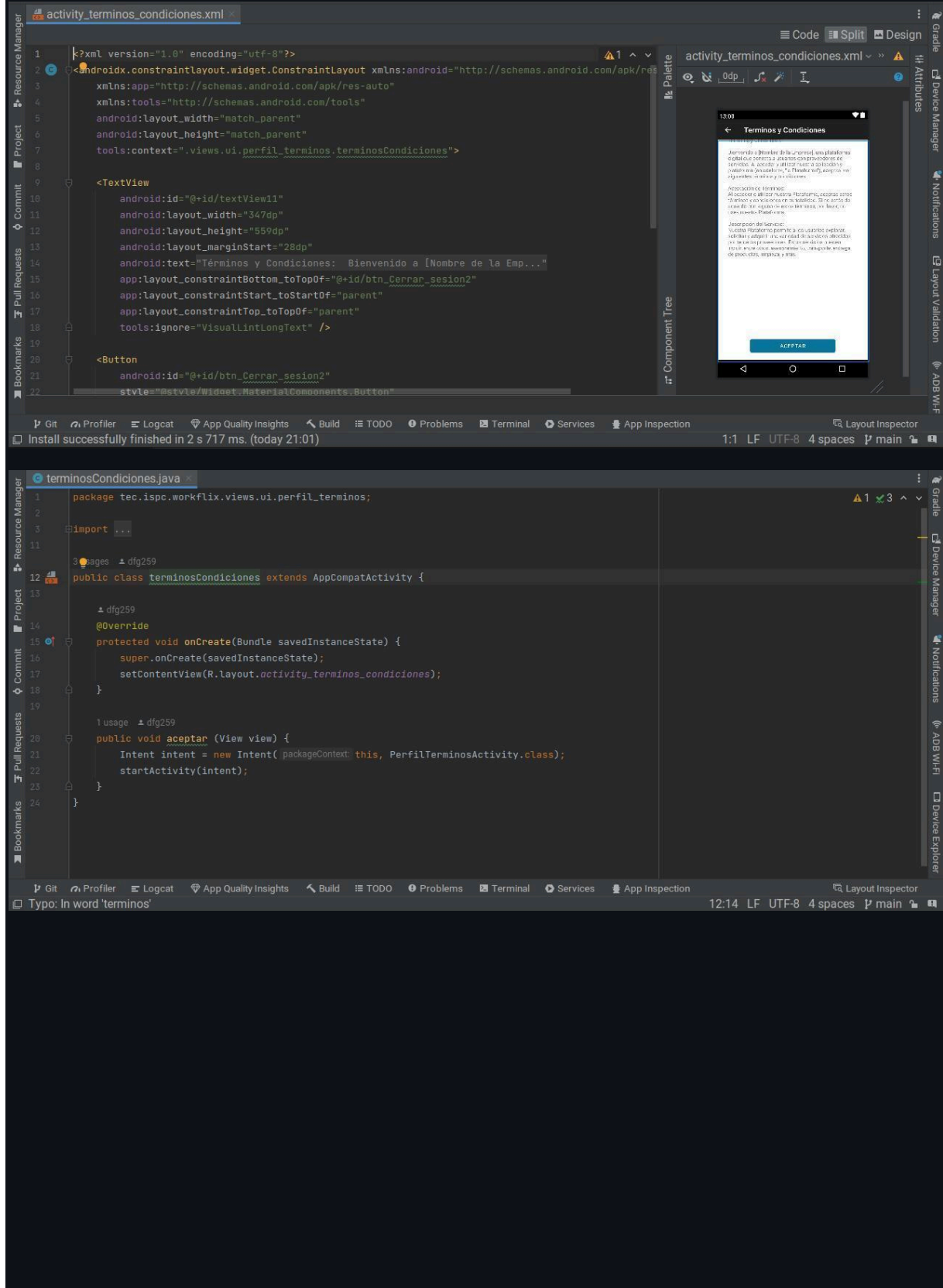
The right sidebar shows a preview of the app's UI, titled "Crear Perfil", which includes a "CAMBIA FOTO" button, a "Nombre" input field, an "Apellido" input field, a "Dominio" input field, a "Correo electrónico" input field, and three buttons: "ACTUALIZAR PERFIL" (green), "ELIMINAR CUENTA" (red), and "CERRAR SESION" (blue).

The bottom status bar indicates the app was installed successfully in 2 s 717 ms. (today 21:01). The bottom right corner shows the status bar with "5:40 LF UTF-8 4 spaces P main".

PERFIL TÉRMINOS



PERFIL TÉRMINOS Y CONDICIONES



The image displays the Android Studio IDE with two main panels showing the implementation of a 'Terminos y Condiciones' (Terms and Conditions) screen.

Top Panel: activity_terminos_condiciones.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3  xmlns:tools="http://schemas.android.com/tools"
4  android:layout_width="match_parent"
5  android:layout_height="match_parent"
6  tools:context=".views.ui.perfil_terminos.terminosCondiciones">
7
8
9  <TextView
10     android:id="@+id/textView11"
11     android:layout_width="347dp"
12     android:layout_height="559dp"
13     android:layout_marginStart="28dp"
14     android:text="Términos y Condiciones: Bienvenido a [Nombre de La Emp..."
15     app:layout_constraintBottom_toTopOf="@+id/btn_Cerrar_sesion2"
16     app:layout_constraintStart_toStartOf="parent"
17     app:layout_constraintTop_toTopOf="parent"
18     tools:ignore="VisualLintLongText" />
19
20 <Button
21     android:id="@+id/btn_Cerrar_sesion2"
22     style="@style/Widget.MaterialComponents.Button"
  
```

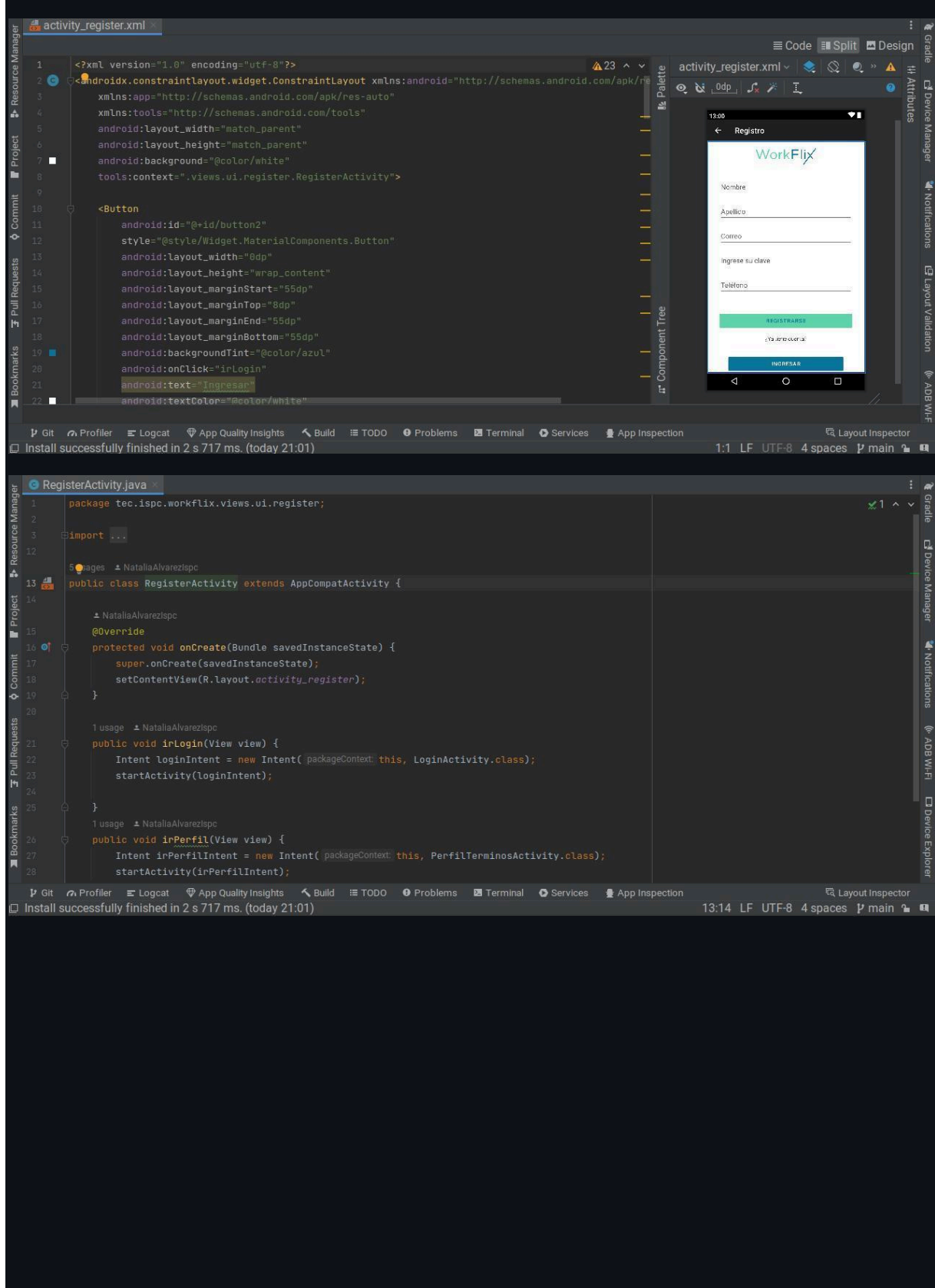
Bottom Panel: terminosCondiciones.java

```

1  package tec.ispc.workflix.views.ui.perfil_terminos;
2
3  import ...
4
5  public class terminosCondiciones extends AppCompatActivity {
6
7      @Override
8      protected void onCreate(Bundle savedInstanceState) {
9          super.onCreate(savedInstanceState);
10         setContentView(R.layout.activity_terminos_condiciones);
11     }
12
13     public void aceptar (View view) {
14         Intent intent = new Intent( packageContext this, PerfilTerminosActivity.class);
15         startActivity(intent);
16     }
17 }
  
```

The interface shows a mobile screen titled 'Terminos y Condiciones' with a large text area for the terms and a blue 'Aceptar' button at the bottom.

REGISTRARSE



RESTABLECER

The screenshot displays the Android Studio interface for the 'Restablecer' (Reset) screen. The top editor shows the XML layout file 'activity_restablecer.xml' with the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    tools:context=".views.ui.restablecer.RestablecerActivity">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="158dp"
        android:layout_height="48dp"
        android:layout_marginTop="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/logo_workflix_1" />
    <EditText
        android:id="@+id/editTextTextEmailAddress"
        ...
    </EditText>
</androidx.constraintlayout.widget.ConstraintLayout>
```

The bottom editor shows the Java code for 'RestablecerActivity.java':

```
package tec.ispc.workflix.views.ui.restablecer;

import ...

4 usages: NataliaAlvarezIspc
public class RestablecerActivity extends AppCompatActivity {
    15
    16 @Override
    17 protected void onCreate(Bundle savedInstanceState) {
    18     super.onCreate(savedInstanceState);
    19     setContentView(R.layout.activity_restablecer);
    20 }
    21 2 usages: NataliaAlvarezIspc
    22 public void irIngreso(View view) {
    23     Intent intent = new Intent(packageContext.this, LoginActivity.class);
    24     startActivity(intent);
    25 }
}
```

The right sidebar shows a preview of the 'Restablecer' screen, which displays the 'WorkFlix' logo, a login form with fields for 'Ingresar su Correo' and 'Ingresar su nueva clave', and buttons for 'CAMBIAR CLAVE' and 'VOLVER'.

