

Reporte Reddit

Introducción	2
Desempeño	2
Tiempos de instalación y corrido en distintos dispositivos	15
Conectividad eventual	16
Seguridad	17
Análisis del código	20
Librerías encontradas	21
Buenas prácticas	23
Malas prácticas	25
Estrategias de almacenamiento	25
Shared preferences	25
Manejo de caché	26
UI	27
Overdraw	27
Diseño	32
Conclusiones	33

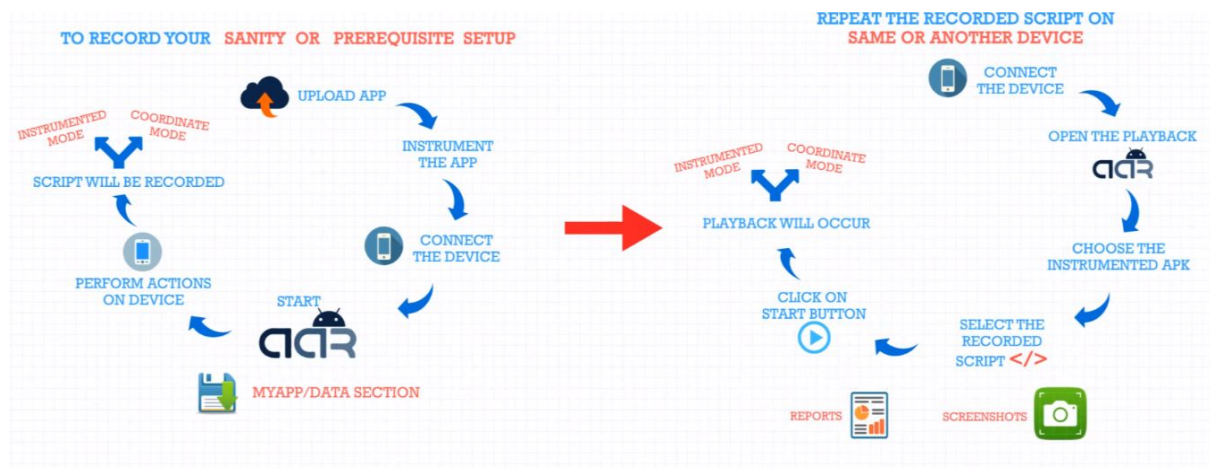
Introducción

Reddit es una aplicación para ver y compartir contenido web. Este contenido puede tener votos positivos o negativos según la percepción de los usuarios. La idea es generar discusión en cuanto a los contenidos, por lo cual se pueden comentar o compartir los posts.

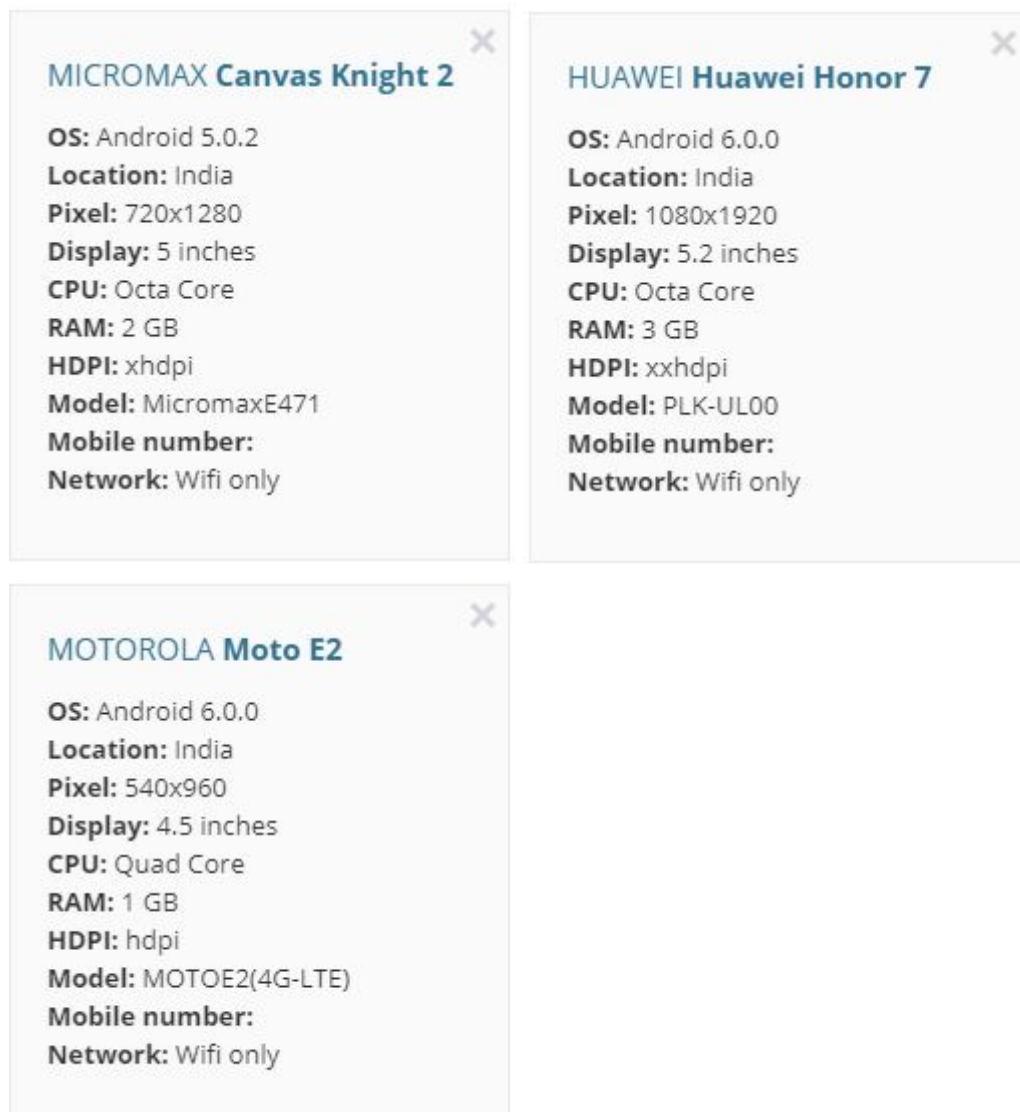
Desempeño

Para medir el desempeño de la aplicación se hizo uso de la versión gratuita de la aplicación web pcloudy. Para evitar variación de tiempos entre actividades producto de la intervención humana en una prueba manual. Se automatizó el flujo un script, se almacenó y posteriormente se ejecutó en los dispositivos móviles. De antemano, debido a que los dispositivos siempre están conectados a una fuente de energía, se descarta el análisis de consumo de batería. Sin embargo, observando las mediciones de la simulación se calculó que la aplicación consumía entre en 12% - 15% de la energía del dispositivo.

El proceso para la creación y uso del script se presenta a continuación:



Las especificaciones de los dispositivos, proporcionados para la herramienta, donde fue posible ejecutar los scripts (android version minima 5.0, gratuito) son:



Se definieron tres flujos típicos de navegación de acuerdo a lo indagado a usuarios universitarios de la Universidad de los Andes:

Flujo Uno: El usuario no hace login en la aplicación. Se dirige a la sección Public donde se muestran los post más populares (publicación reciente con gran número de comentarios y likes). El usuario selecciona un post y hace scroll para leer los comentarios de la aplicación.

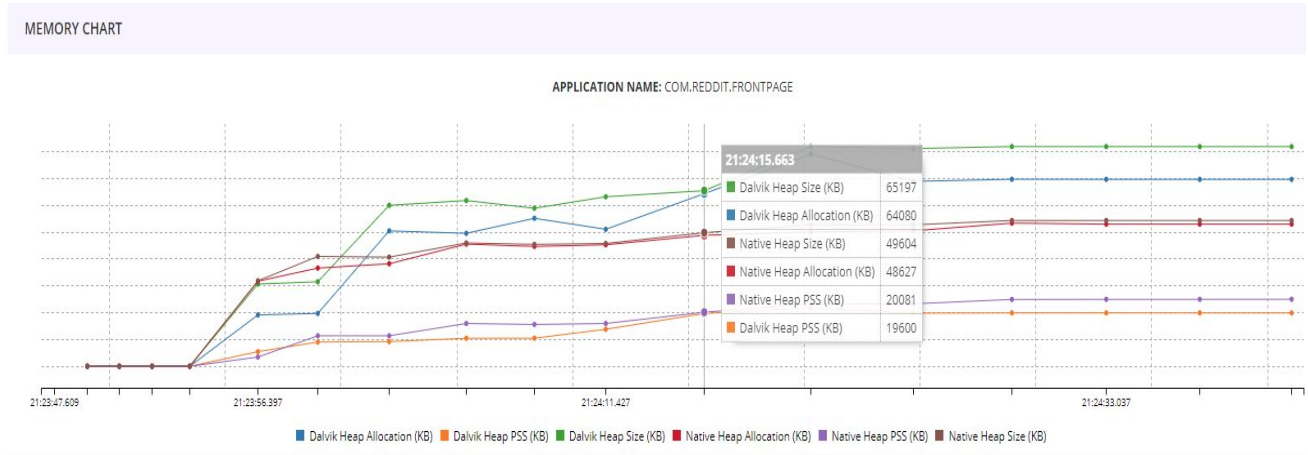
Nota: en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_1/](#) del repositorio se encuentra las capturas de pantalla del flujo paso a paso.

Por otra parte, se realizó una captura en video de la simulación automatizada de este flujo en el dispositivo Moto E2. este video se encuentra en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_1/Flujo1_VIDEO.mp4](#) del repositorio

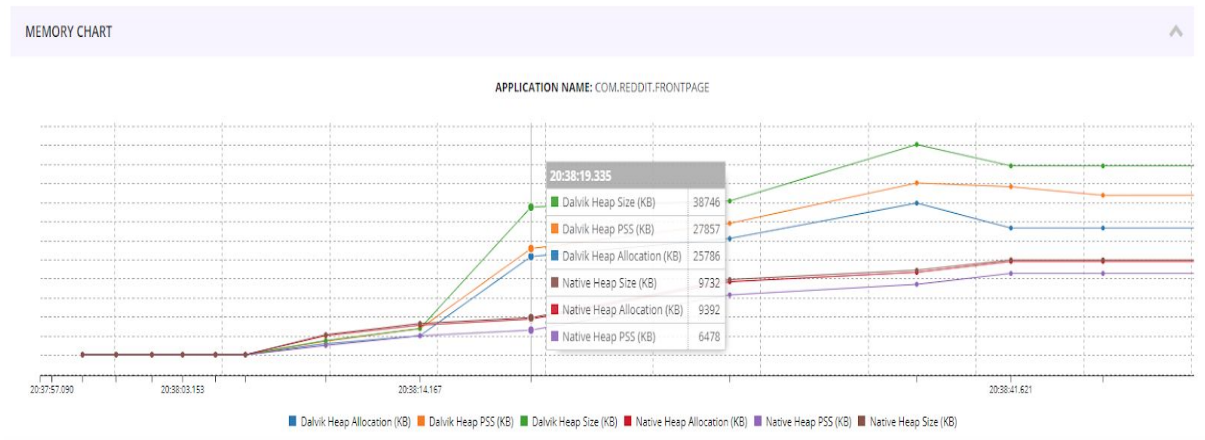
A continuación se presentan las capturas de pantalla del consumo de memoria, cpu y red en los dispositivos simulados. Para observar a mayor resolución las imágenes dirigirse a la dirección [ReporteReddit/CapturaFlujos/Flujo_1/](#) del repositorio

Memoria:

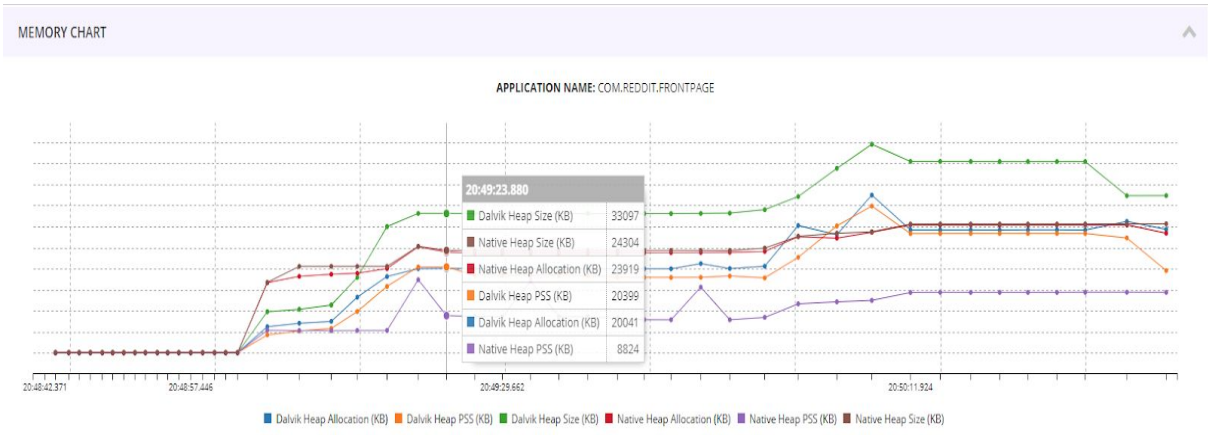
Respecto al consumo de memoria se puede ver como se incrementa una vez se puede ver como el primer pico corresponde a traer los post más populares. Después, el segundo pico corresponde a traer toda la información de un post. Más adelante en el reporte se demuestra que al traer los comentarios de un post se está creando más de un objeto en un ciclo, lo cual causa un gran consumo de memoria



HUAWEI HONOR 7

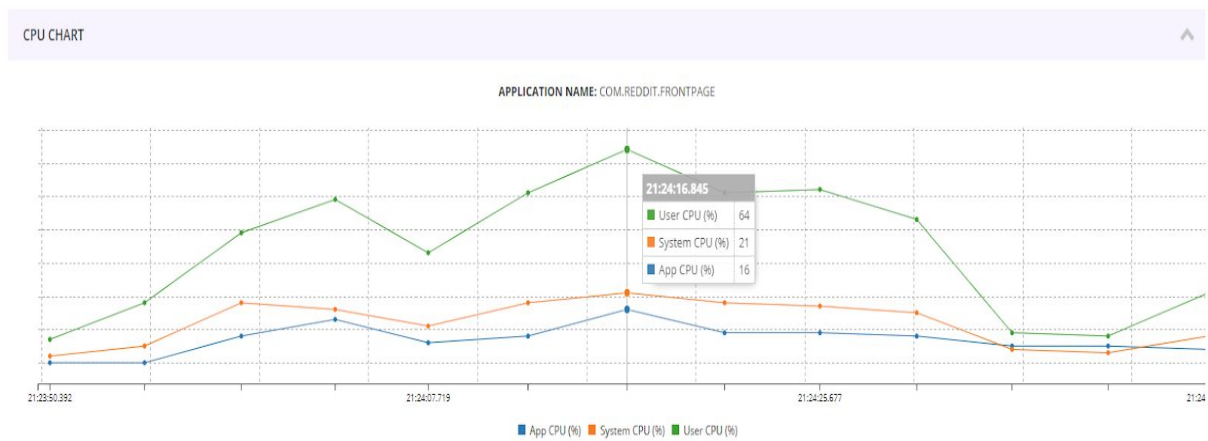


MICROMAX KNIGHT 2

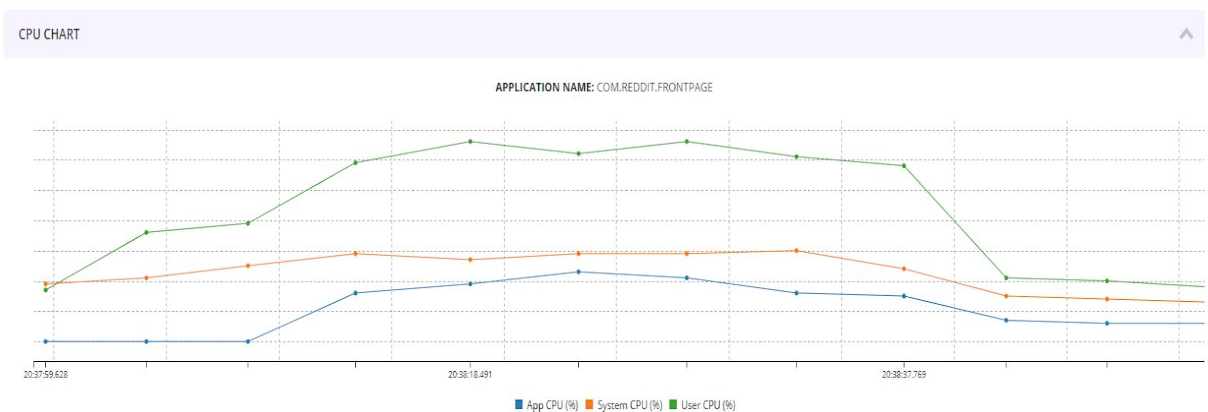


MOTO E2

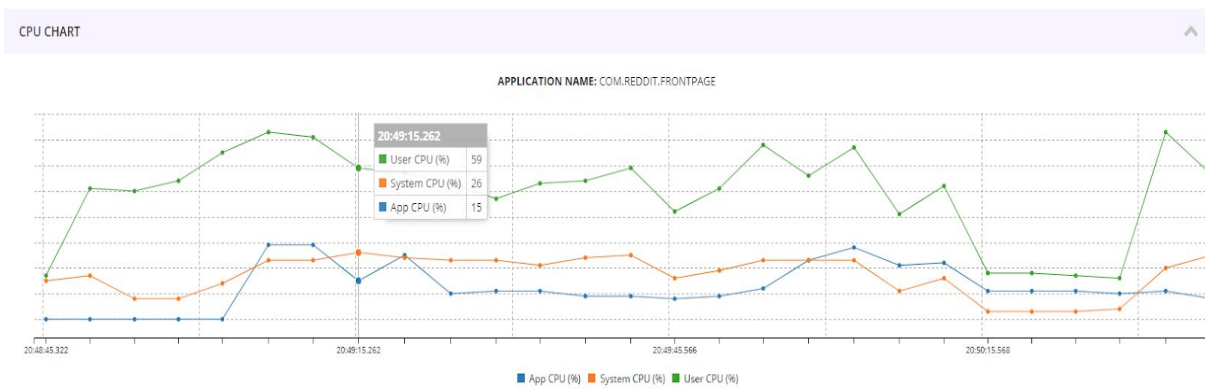
CPU:



HUAWEI HONOR 7



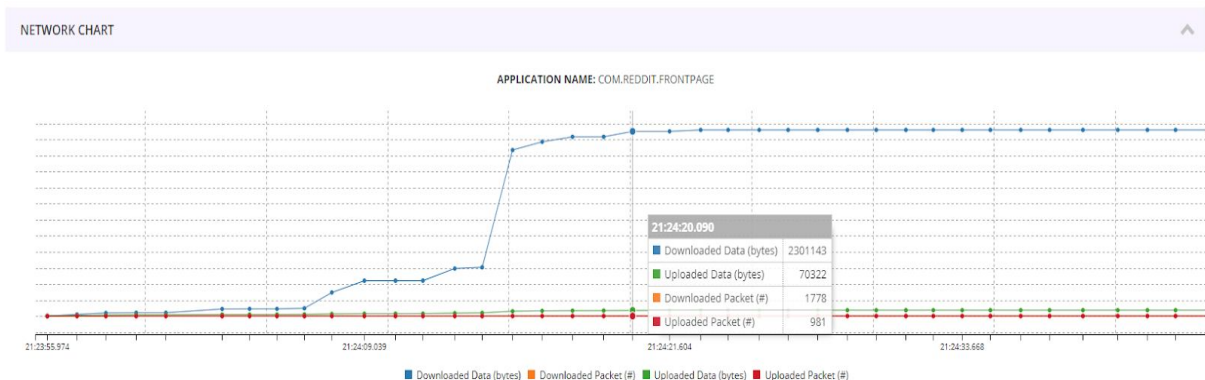
MICROMAX KNIGHT 2



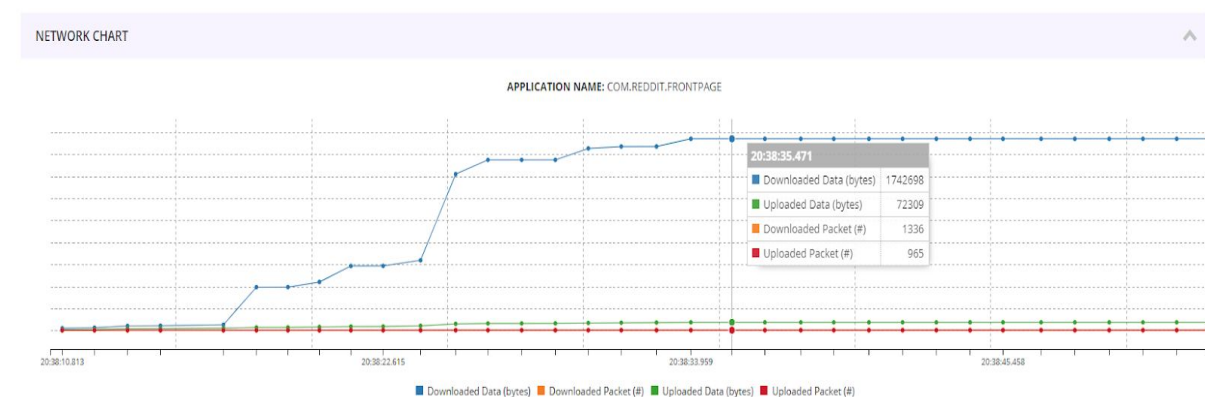
MOTO E2

Red

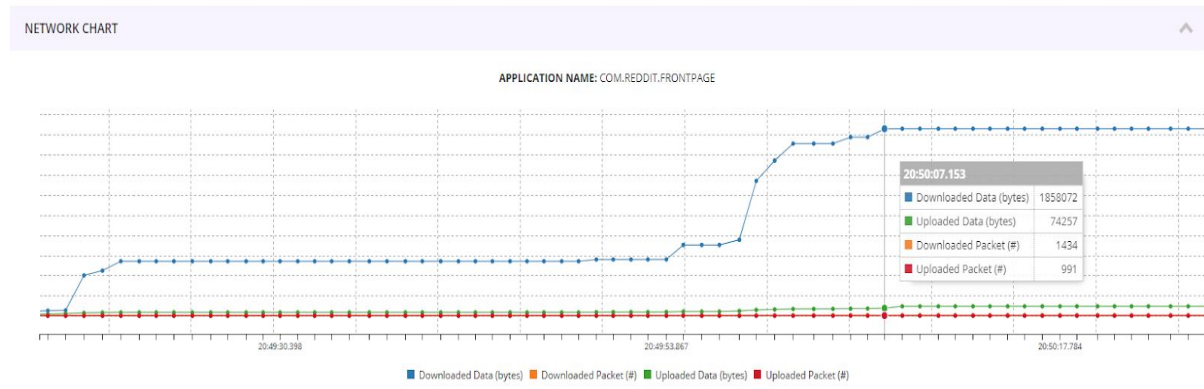
Respecto al consumo de red, vemos un leve incremento al momento de cargar la pantalla principal, ya que (sin necesidad de hacer tap en la pestaña popular), se está trayendo la información de estos posts. Después, el pico se presenta porque se hace scroll en los posts, trayendo las imágenes multimedia de cada post .



HUAWEI HONOR 7



MICROMAX KNIGHT 2



MOTO E2

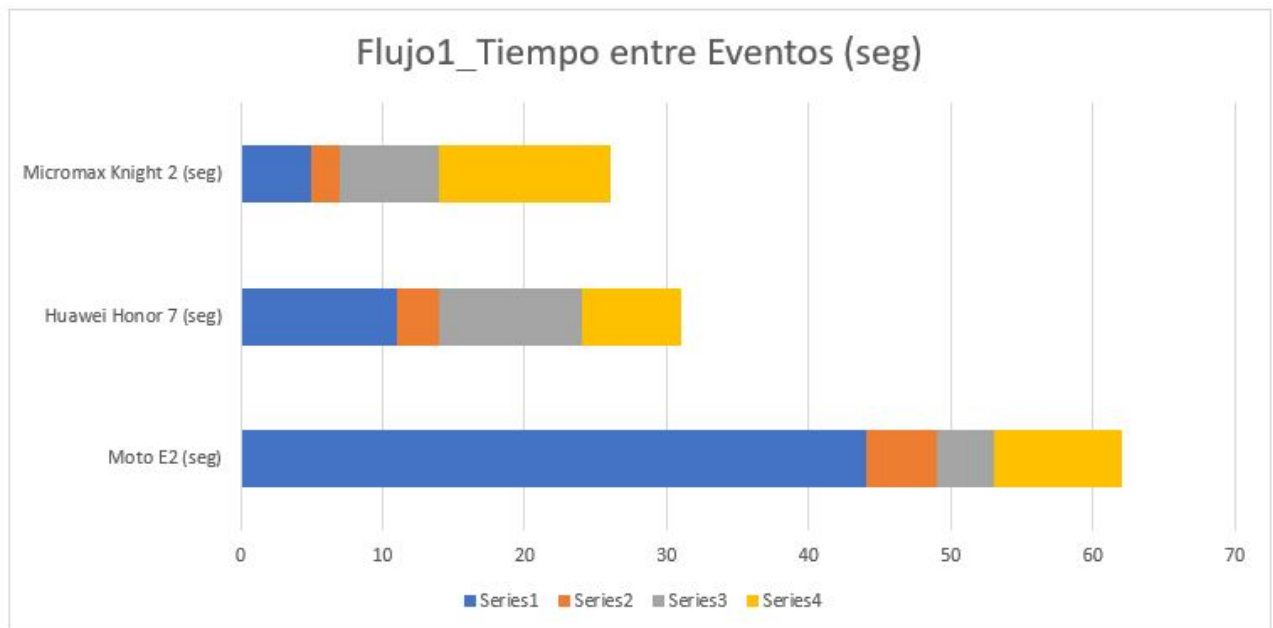
Tiempo entre eventos

En la carpeta **ReporteReddit/CapturaFlujos/** se presenta un reporte HTML de los tiempos de respuesta de cada dispositivo entre el paso un evento de simulación a otro. Cada archivo HTML cuenta con una captura automática de cada evento o en su defecto cada evento está especificado en el archivo de automatización .rec que se encuentra en la dirección **ReporteReddit/VideosFlujosAutomatizados/Flujo_1/Flujo1_instruccionesAutomatizadas.rec** del repositorio.

A continuación presentamos una tabla con los tiempos entre cada actividad. Respecto al tiempo inicial del equipo Moto E2 se presenta un alto tiempo de arranque, debido a que es el dispositivo con menor tamaño de ram.

Moto E2 (seg)	Huawei Honor 7 (seg)	Micromax Knight 2 (seg)
44	11	5
5	3	2
4	10	7
9	7	12

En la gráfica, cada serie "X" representa el evento "X"



Flujo 2: El usuario hace login. Se dirige a la sección de mensajes, escribe un texto a un destinatario: ingresa destinatario r/<amigo>, ingresa un asunto y el texto del mensaje. Por último, revisa su bandeja de entrada.

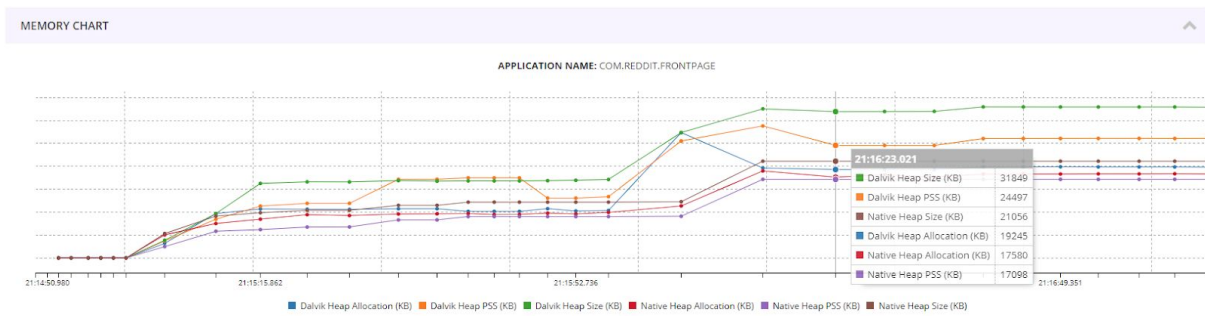
Nota: en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_2/](#) del repositorio se encuentra las capturas de pantalla del flujo paso a paso.

Por otra parte, se realizó una captura en video de la simulación automatizada de este flujo en el dispositivo Moto E2. este video se encuentra en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_2/Flujo2_VIDEO.mp4](#) del repositorio

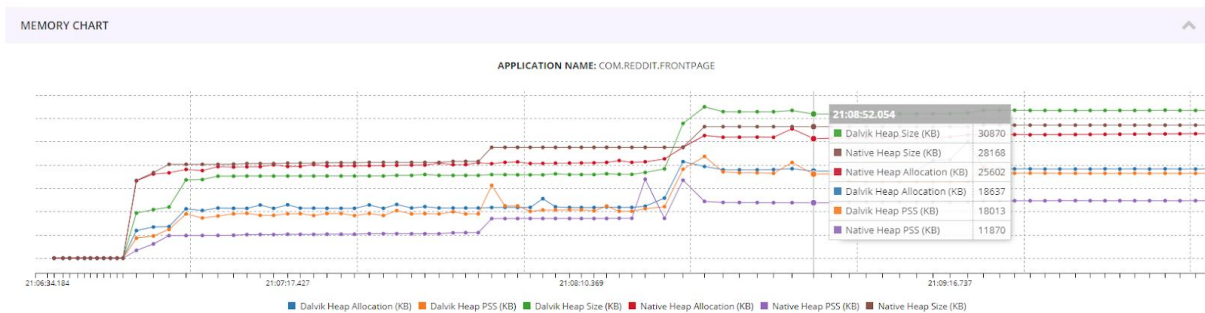
A continuación se presentan las capturas de pantalla del consumo de memoria, cpu y red en los dispositivos simulados. Para observar a mayor resolución las imágenes dirigirse a la dirección [ReporteReddit/CapturaFlujos/Flujo_2/](#) del repositorio

Memoria:

El incremento de memoria en la mitad de la gráfica corresponde a traer los posts correspondientes al usuario logueado (lo cuál se hace en otro thread para no sobrecargar el thread principal. Después, no se presenta mayor aumento debido a que los mensajes en preview no tienen contenido multimedia.



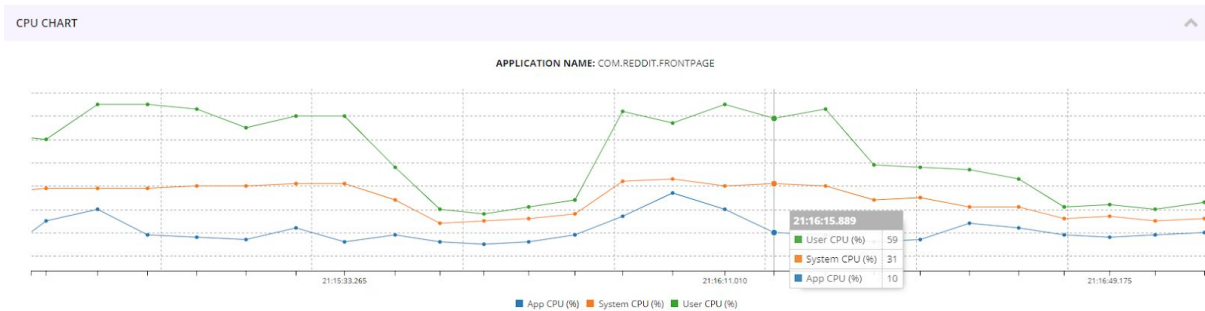
MICROMAX KNIGHT 2



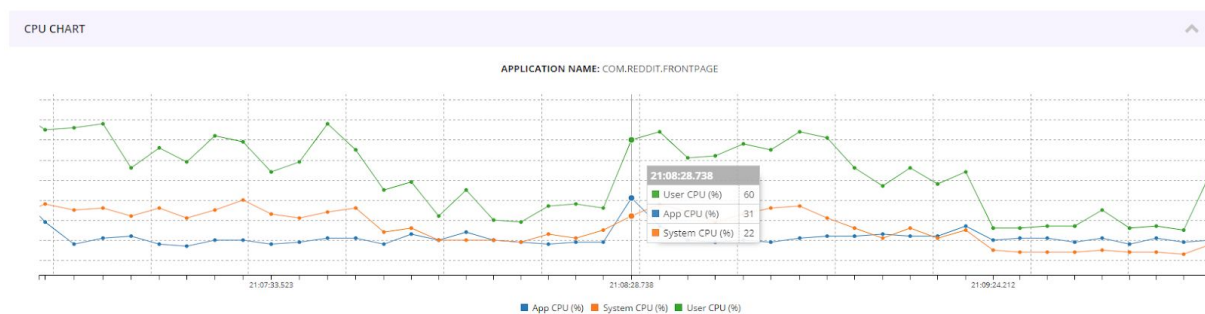
MOTO E2

CPU:

Al igual que en el uso de memoria, el consumo de cpu aumenta en la mitad de la gráfica debido al login exitoso por parte del usuario. Por otro lado, el consumo de cpu va disminuyendo debido a que en el evento enviar mensaje se están ejecutando menos procesos en threads secundarios.



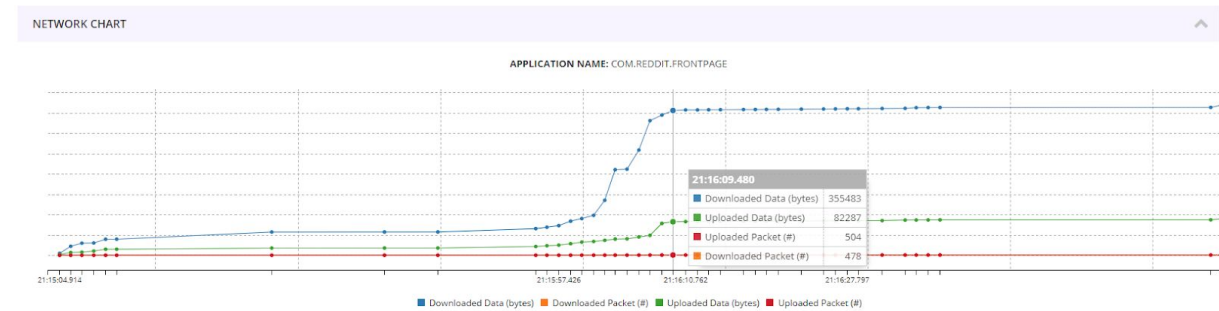
MICROMAX KNIGHT 2



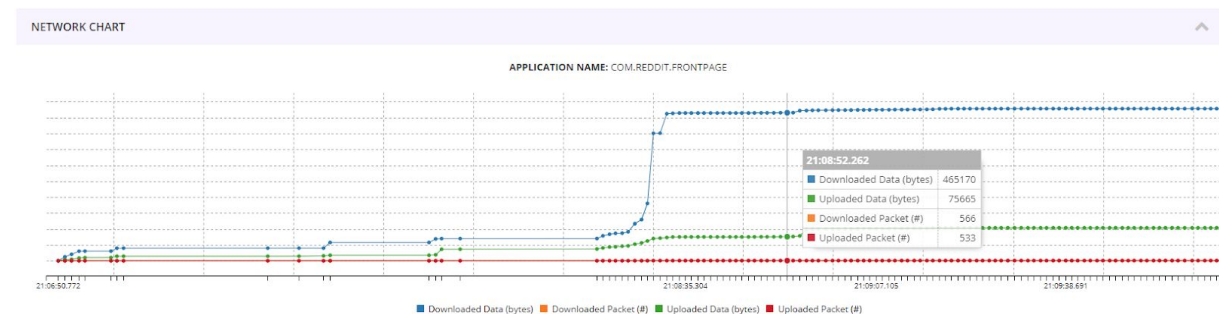
MOTO E2

Red:

Por último, el uso de red, como es esperado, tiene un pico cuando el usuario realiza el login y se hacen traer los posts en un thread secundario



MICROMAX KNIGHT 2



MOTO E2

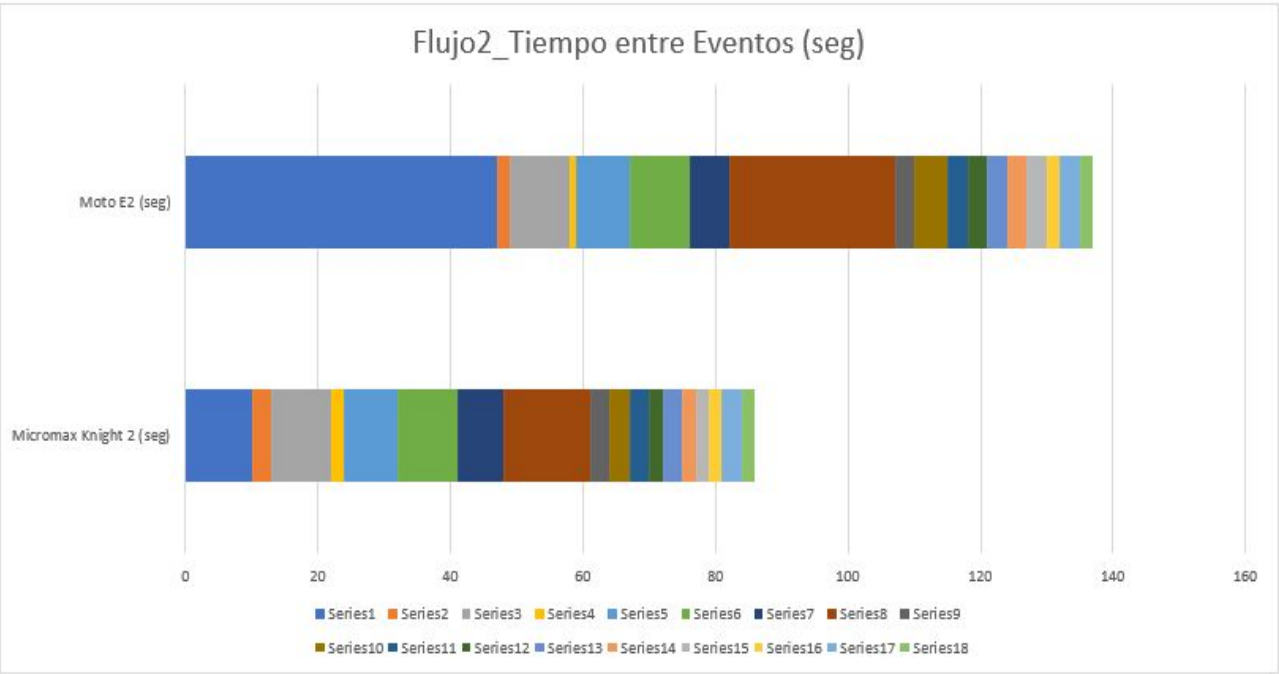
Tiempo entre eventos

En la carpeta **ReporteReddit/CapturaFlujos/** se presenta un reporte HTML de los tiempos de respuesta de cada dispositivo entre el paso un evento de simulación a otro. Cada archivo HTML cuenta con una captura automática de cada evento o en su defecto cada evento está especificado en el archivo de automatización .rec que se encuentra en la dirección **ReporteReddit/VideosFlujosAutomatizados/Flujo_1/Flujo1_instruccionesAutomatizadas.rec** del repositorio.

El tiempo en la primera actividad es mucho mayor en el dispositivo Moto E2 debido a que sus especificaciones de ram son inferiores comparadas con MicroMax Knight 2. Después, los tiempos 13 y 25 seg para cada dispositivo corresponden a la actividad home donde se visualizan los posts. Como fue demostrado anteriormente, se usan más recursos y se toma más en esta parte de la ejecución de la prueba. Mientras que, en comparación, enviar un mensaje consume menos cpu y memoria.

Micromax Knight 2 (seg)	Moto E2 (seg)
10	47
3	2
9	9
2	1
8	8
9	9
7	6
13	25
3	3
3	5
3	3
2	3
3	3
2	3
2	3
2	2
3	3
2	2

En la gráfica, cada serie "X" representa el evento "X"



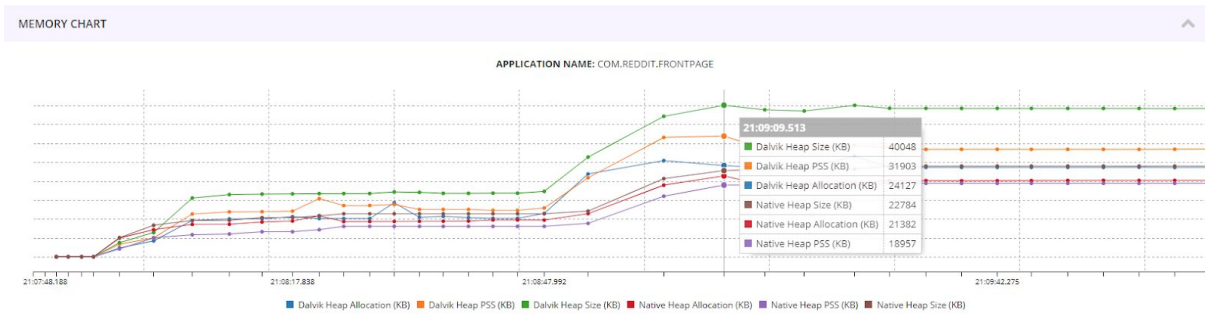
Flujo 3: El usuario hace login. Se dirige a los subreddits a los que está inscrito y selecciona uno de estos foros. Posteriormente, selecciona el botón de publicación de texto e ingresa el nombre del post y su contenido. Finalmente se da click en post.

Nota: en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_3/](#) del repositorio se encuentra las capturas de pantalla del flujo paso a paso.

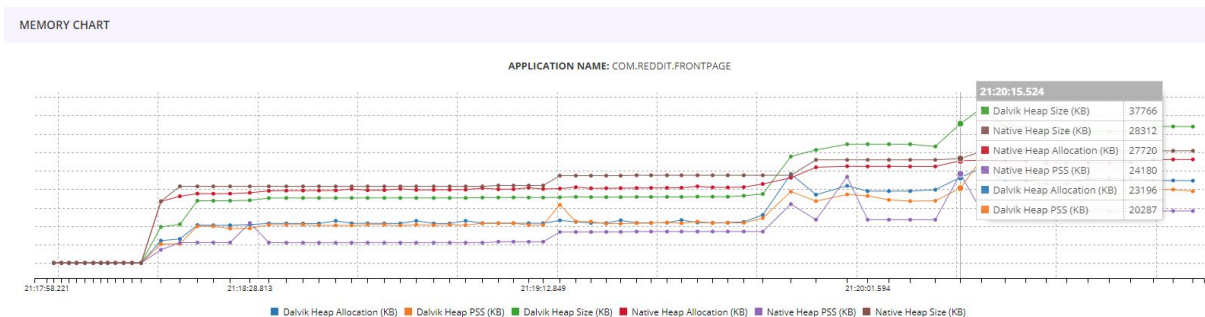
Por otra parte, se realizó una captura en video de la simulación automatizada de este flujo en el dispositivo Moto E2. este video se encuentra en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_3/Flujo3_VIDEO.mp4](#) del repositorio

A continuación se presentan las capturas de pantalla del consumo de memoria, cpu y red en los dispositivos simulados. Para observar a mayor resolución las imágenes dirigirse a la dirección [ReporteReddit/CapturaFlujos/Flujo_3/](#) del repositorio

Memoria

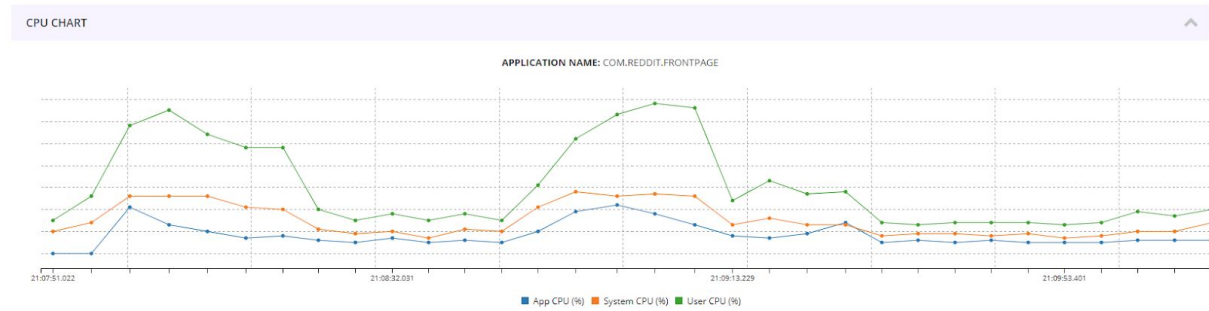


MICROMAX KNIGHT 2

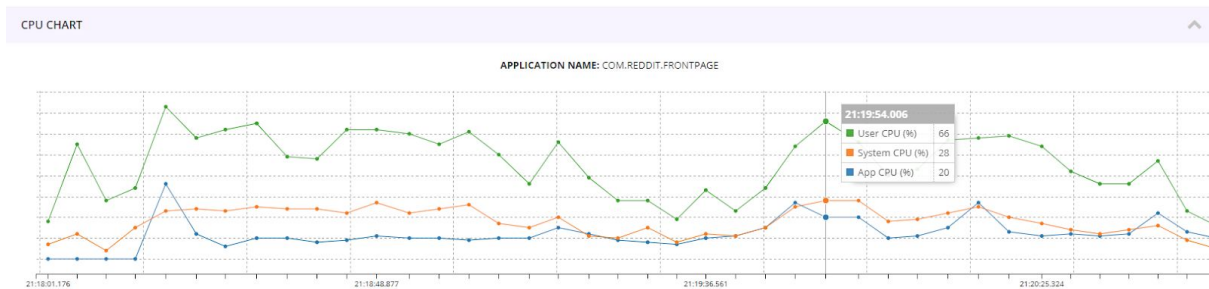


MOTO E2

CPU



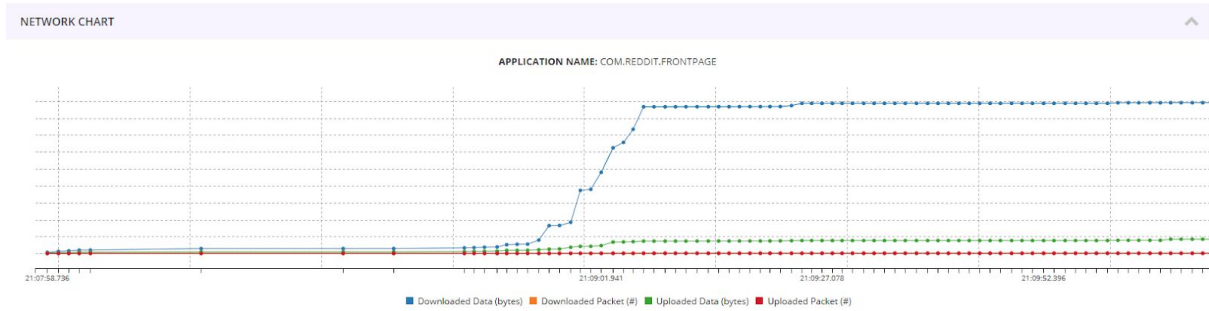
MICROMAX KNIGHT 2



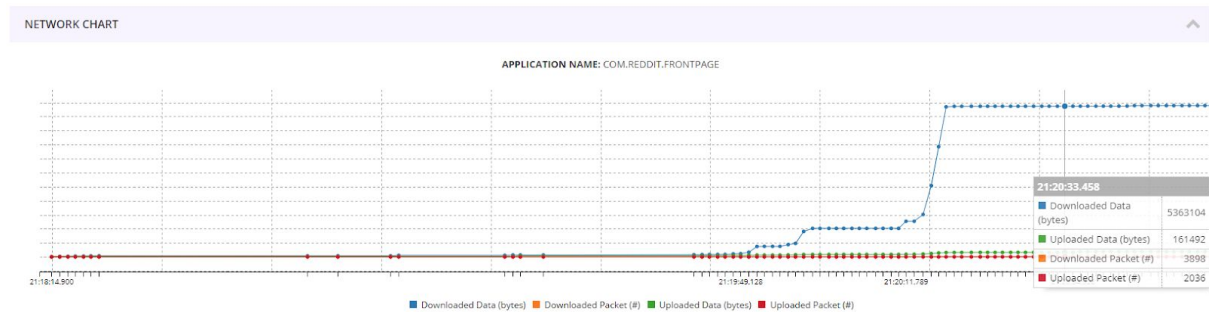
MOTO E2

Red

Por último, el uso de red, como es esperado, tiene un pico cuando el usuario realiza el login y se hacen traer los posts en un thread secundario



MICROMAX KNIGHT 2



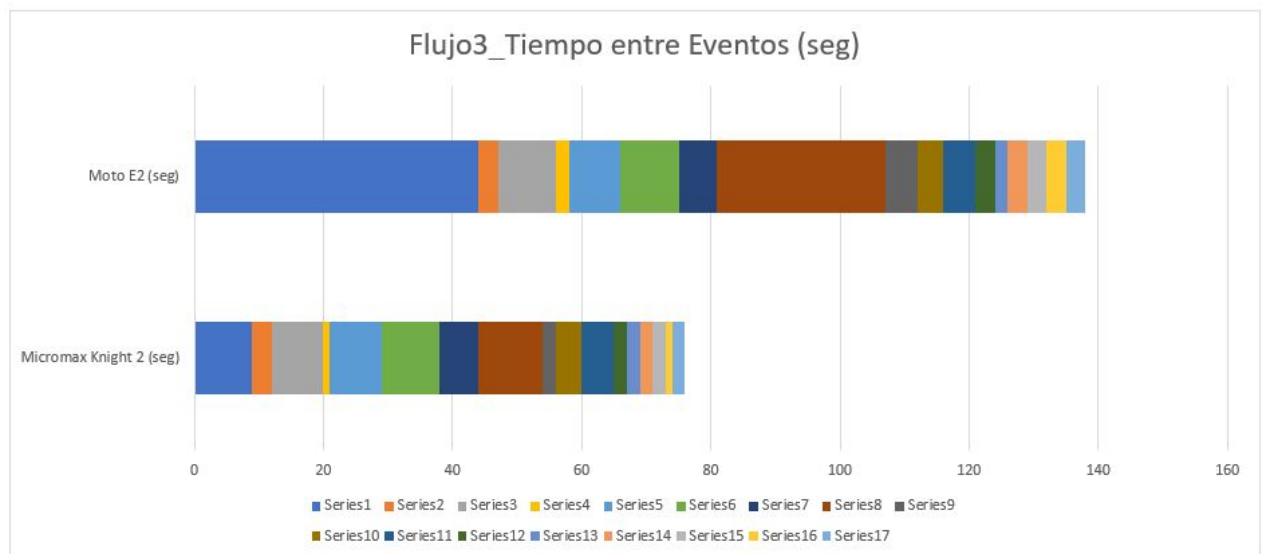
MOTO E2

Tiempo entre eventos

En la carpeta **ReporteReddit/CapturaFlujos/** se presenta un reporte HTML de los tiempos de respuesta de cada dispositivo entre el paso un evento de simulación a otro. Cada archivo HTML cuenta con una captura automática de cada evento o en su defecto cada evento está especificado en el archivo de automatización .rec que se encuentra en la dirección **ReporteReddit/VideosFlujosAutomatizados/Flujo_1/Flujo1_instruccionesAutomatizadas.rec** del repositorio.

Micromax Knight 2 (seg)	Moto E2 (seg)
9	44
3	3
8	9
1	2
8	8
9	9
6	6
10	26
2	5
4	4
5	5
2	3
2	2
2	3
2	3
1	3
2	3

En la gráfica, cada serie "X" representa el evento "X"



Tiempos de instalación y corrido en distintos dispositivos

Los tiempos de instalación, launch y corrido de la aplicación se calcularon haciendo uso de la versión gratuita de la aplicación web Bitbar (bitbar.com)

	LG Nexus 5	LG Nexus 5 D820
Procesador	Qualcomm Snapdragon 800	Qualcomm MSM8974 Snapdragon 800
Nucleos	4 (Krait 400)	4 (Krait 400)
Frecuencia	2,26 GHz	2.3 GHz Krait 400
RAM	2 GB	2 GB
Memoria Interna	16 GB	16 GB
Android	6,0,1	5

Device Statuses (2)				Search devices	
<input checked="" type="checkbox"/> Device	Tests	Time Bar	Dev...		
<input type="checkbox"/> LG Google Nexus 5 6.0.1 (...) Android 6.0.1	100% 1/1	43s 1m 31s of 2m 14s	Succeeded		
<input type="checkbox"/> LG Google Nexus 5 D820 ... Android 5.0	100% 1/1	48s 4s 1m 24s of 2m 16s	Succeeded		
Legend Installing Launching Running					

Para ambos dispositivos se observan tiempos similares. Sin embargo, el Nexus 5 D820 tiene 4 segundos de launching mientras que el otro dispositivo no lo hace.

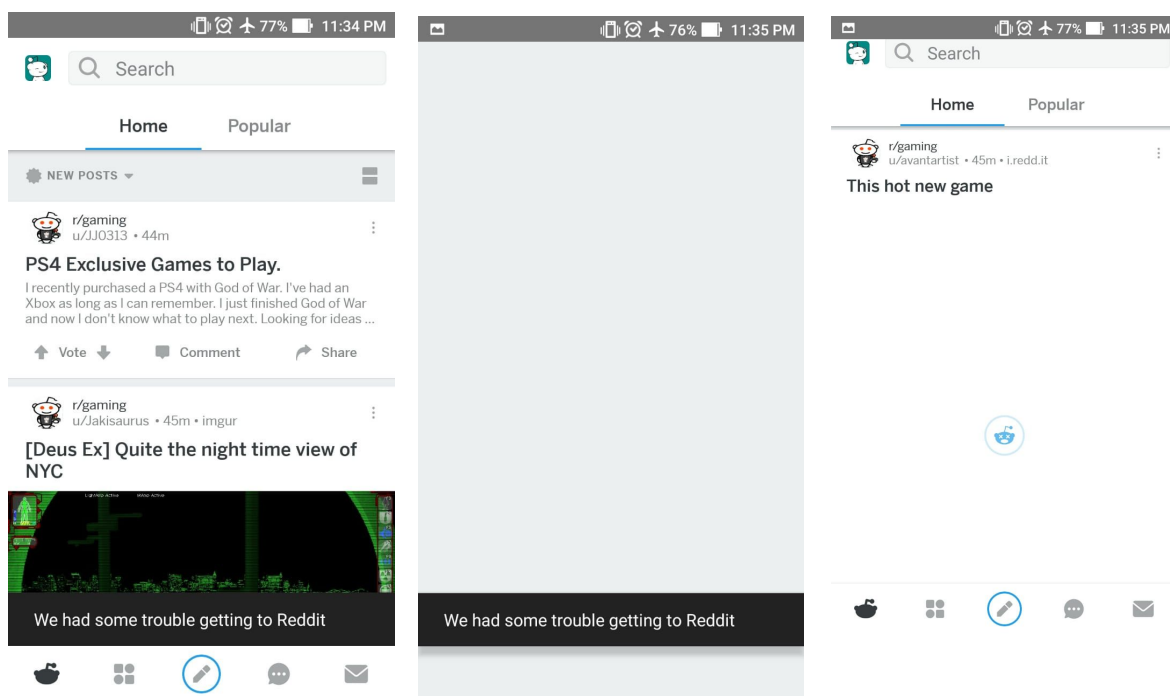
Conectividad eventual

Los mensajes que se muestran para conectividad eventual son poco explícitos o inexistentes.

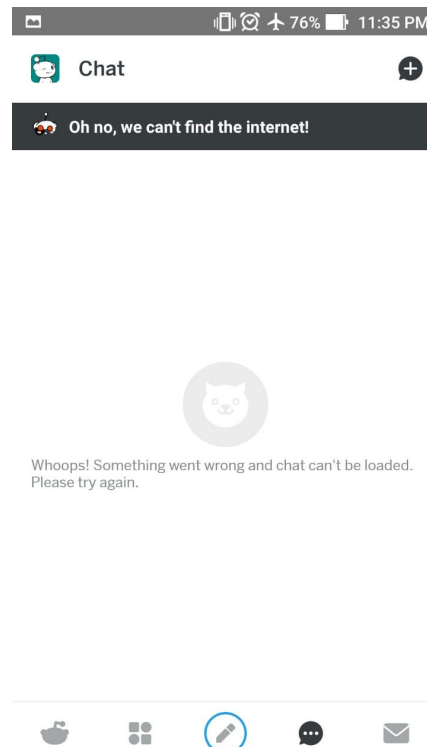
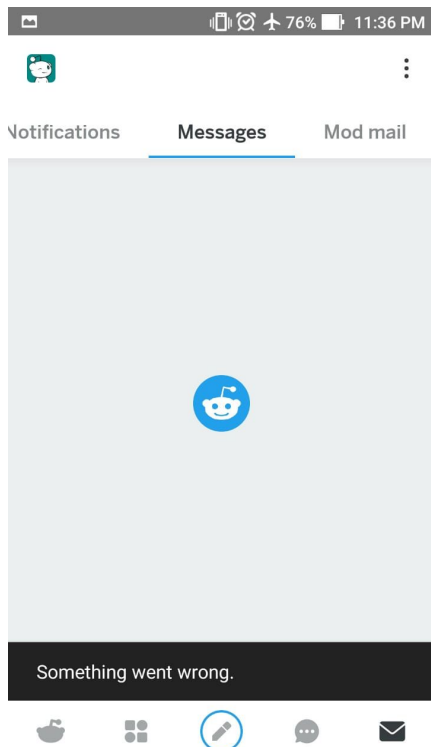
En la primera imagen se tenía wifi activado en la lista de mensajes, sin embargo, se desactivo antes de entrar a un mensaje en particular.

La primera prueba que se realizó fue cerrar la aplicación, desactivar el acceso a internet e iniciar la aplicación:

En la imagen de la izquierda, se puede observar que se cargan posts que están guardados en caché o en disco pero el mensaje “*We had some trouble getting to Reddit*” no da a entender que el usuario no tiene internet. En la imagen del centro, se muestra el mismo mensaje de error y la pantalla completamente blanca. Finalmente, en la imagen de la derecha se ve como se muestra un post del cual no se pudo obtener su imagen.



Por otro lado, también se navegó la pestaña de mensajes y obtuvimos el mensaje *"Something went wrong"*. Algo particular es que en la vista de chats, sí se obtuvo un mensaje *"Oh no, we can't find the internet!"*, el cuál es más explícito para el usuario



Seguridad

Para probar la seguridad de la aplicación se decidió usar la herramienta MobSF¹. Detectamos que la aplicación usa la función Hash de Java y no es seguro implementarla.

¹ <https://github.com/MobSF/Mobile-Security-Framework-MobSF>

ISSUE	SEVERITY	FILES
This App uses Java Hash Code. It's a weak hash function and should never be used in Secure Crypto Implementation.	high	okhttp3/Handshake.java okhttp3/HttpUrl.java okhttp3/CertificatePinner.java okhttp3/Challenge.java okhttp3/TlsVersion.java okhttp3/Cookie.java okhttp3/Address.java okhttp3/MediaType.java okhttp3/Route.java okhttp3/internal/http2/Header.java okhttp3/internal/platform/AndroidPlatform.java okhttp3/internal/tls/BasicCertificateChainCleaner.java okhttp3/internal/tls/BasicTrustRootIndex.java de/greenrobot/event/Subscription.java de/greenrobot/event/SubscriberMethod.java com/jakewharton/rxbinding2/widget/AutoValue_TextViewAfterTextChangeEvent.java com/airbnb/deeplinkdispatch/DeepLinkUri.java com/sendbird/android/BaseMessage.java com/sendbird/android/SendBird.java com/sendbird/android/shadow/okhttp3/Handshake.java com/sendbird/android/shadow/okhttp3/HttpUrl.java com/sendbird/android/shadow/okhttp3/CertificatePinner.java

También se encontró que se usaba un generador de números aleatorios inseguro

The App uses an insecure Random Number Generator.	high	okhttp3/OkHttpClient.java okhttp3/internal/ws/RealWebSocket.java okhttp3/internal/ws/WebSocketWriter.java com/sendbird/android/WSClient.java com/sendbird/android/shadow/okhttp3/internal/ws/RealWebSocket.java com/sendbird/android/shadow/okhttp3/internal/ws/WebSocketWriter.java com/crashlytics/android/answers/RandomBackoff.java com/reddit/frontpage/util/NotificationUtil.java com/reddit/datalibrary/social/data/session/ChatConnectionManager.java org/jsoup/helper/DataUtil.java kotlin/reflect/jvm/internal/impl/utils/SmartList.java kotlin/reflect/jvm/internal/impl/protobuf/LazyStringArrayList.java kotlin/reflect/jvm/internal/impl/protobuf/UnmodifiableLazyStringList.java kotlin/collections/ArraysKt___ArraysKt.java kotlin/collections/EmptyList.java kotlin/collections/CollectionsKt__MutableCollectionsKt.java kotlin/collections/RingBuffer.java kotlin/collections/AbstractList.java kotlin/collections/CollectionsKt___CollectionsKt.java io/reactivex/internal/util/VolatileSizeArrayList.java net/hockeyapp/android/utis/SimpleMultipartEntity.java okio/Options.java
---	------	--

El uso de md5 tanto en las librerías como en el package de reddit no es recomendado

MD5 is a weak hash known to have hash collisions.	high	com/danikula/videocache/ProxyCacheUtils.java com/instabug/library/util/b.java com/reddit/frontpage/util/FileUtil.java com/reddit/social/util/FileUtils.java org/jcodec/common/tools/MD5.java io/branch/indexing/ContentDiscoverer.java
---	------	---

Además, se encontraron actividades con intent filter y exported=true, lo que indica que pueden ser accedidas por cualquier otra aplicación en el dispositivo

Activity (com.reddit.frontpage.MainActivity) is not Protected. An intent-filter exists.	high
Activity (com.reddit.frontpage.redditauth_private.ui.AuthActivity) is not Protected. [android:exported=true]	high
Activity (com.reddit.frontpage.RedditDeepLinkActivity) is not Protected. An intent-filter exists.	high
Activity (com.reddit.frontpage.BranchLinkActivity) is not Protected. An intent-filter exists.	high

El hecho de tener servicios y Broadcast Receivers representa un alto riesgo de seguridad ya que otras aplicaciones pueden tener acceso a ellos

Service (com.reddit.frontpage.redditauth_private.account.AuthService) is not Protected. [android:exported=true]	high
Service (com.reddit.frontpage.sync.UserDataSyncService) is not Protected. [android:exported=true]	high
Service (com.reddit.frontpage.sync.AppDataSyncService) is not Protected. [android:exported=true]	high
Broadcast Receiver (com.reddit.frontpage.notifications.LocalNotificationReceiver) is not Protected. An intent-filter exists.	high
Broadcast Receiver (io.branch.referral.InstallListener) is not Protected. [android:exported=true]	high
Service (com.reddit.frontpage.service.fcm.RedditMessagingService) is not Protected. [android:exported=true]	high

Finalmente, al analizar los permisos en el archivo manifest, se reporta que la aplicación puede leer y escribir datos en memoria externa del celular, manejar cuentas de usuario y mantener la pantalla encendida lo que aumenta el consumo de batería y lo deja expuesto a posibles ataques.

android.permission.WRITE_SYNC_SETTINGS	dangerous
android.permission.WAKE_LOCK	dangerous
com.google.android.c2dm.permission.RECEIVE	dangerous
com.reddit.frontpage.permission.C2D_MESSAGE	signature
android.permission.ACCESS_COARSE_LOCATION	dangerous
android.permission.REQUEST_INSTALL_PACKAGES	dangerous
android.permission.INTERNET	dangerous
android.permission.READ_EXTERNAL_STORAGE	dangerous
android.permission.WRITE_EXTERNAL_STORAGE	dangerous
android.permission.AUTHENTICATE_ACCOUNTS	dangerous
android.permission.GET_ACCOUNTS	normal
android.permission.USE_CREDENTIALS	dangerous
android.permission.MANAGE_ACCOUNTS	dangerous
android.permission.READ_SYNC_SETTINGS	normal
android.permission.WRITE_SYNC_SETTINGS	dangerous

Análisis del código

Debido a que Reddit no es una aplicación de código abierto, se decompiló a partir del apk mediante la herramienta <http://www.javadecompilers.com/apk>. Esto arrojó un archivo .zip con el proyecto que fue importado en Android Studio para mayor facilidad visualizando los archivos. Cabe aclarar que el proyecto no puede ser corrido ya que hacen falta archivos como el build.gradle para que el IDE lo pueda compilar y ejecutar en un dispositivo.

En los archivos generados mediante esta técnica, se encontró código java que había sido compilado a partir de Kotlin como lo sugiere la siguiente imagen:

```

1 package com.reddit.frontpage.di.module;
2
3 import com.reddit.common.rx.BackgroundThread;
4 import com.reddit.datalibrary.frontpage.data.feature.account.datasource.local.LocalAccountDataSource;
5 import com.reddit.datalibrary.frontpage.data.feature.account.datasource.remote.RemoteAccountDataSource;
6 import com.reddit.datalibrary.frontpage.data.feature.account.repo.RedditAccountRepository;
7 import com.reddit.frontpage.domain.repository.AccountRepository;
8 import com.reddit.frontpage.domain.usecase.AccountUseCase;
9 import javax.inject.Singleton;
10 import kotlin.Metadata;
11 import kotlin.jvm.internal.Intrinsics;
12 import retrofit2.Retrofit;
13
14 @Metadata(bv = {1, 0, 2}, d1 = {"\u00004\n\u0002\u0001\u0002\u0002\n\u0000\u00010\u00000\n"},
15 /* compiled from: AccountDataModule.kt */
16 public final class AccountDataModule {
17     @Singleton
18     public static RemoteAccountDataSource m2231a(Retrofit retrofit) {
19         Intrinsics.m26847b(retrofit, "client");
20         Object a = retrofit.a(RemoteAccountDataSource.class);
21         Intrinsics.m26843a(a, "client.create(RemoteAccountDataSource::class.java)");
22         return (RemoteAccountDataSource) a;
23     }
24
25     @Singleton
26     public static AccountRepository m2231a(BackgroundThread backgroundThread, RemoteAccountDataSource remoteAccountDataSource) {
27         Intrinsics.m26847b(backgroundThread, "backgroundThread");
28         return new AccountRepository(backgroundThread, remoteAccountDataSource);
29     }
30 }

```

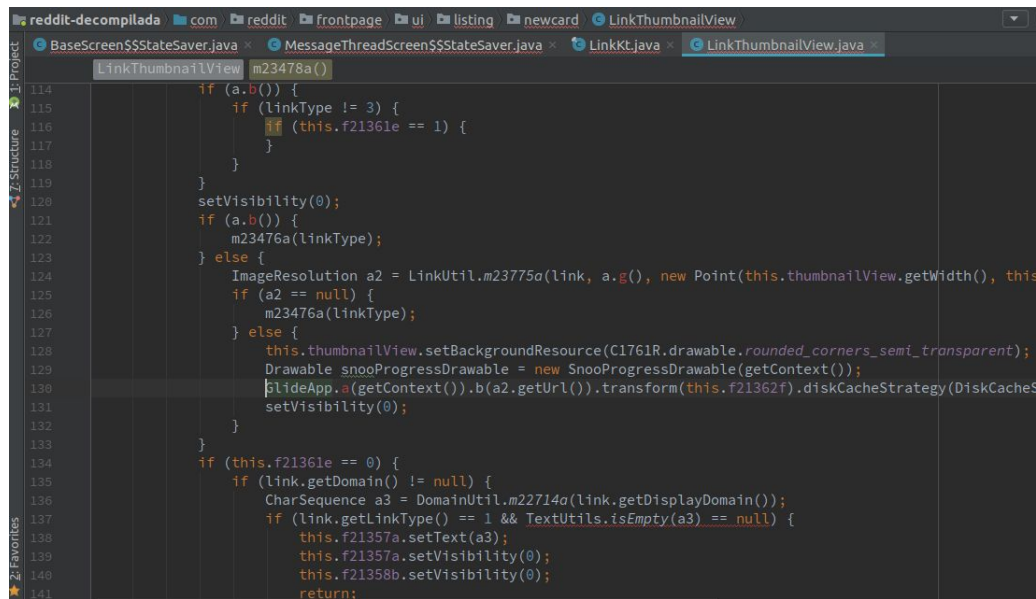
Librerías encontradas

Reddit hace uso de una gran cantidad de librerías, a continuación se explicarán las más relevantes con su correspondiente link al repositorio en github/página oficial.

- Glide

Para mostrar imágenes, thumbnails de videos y gifs. Su principal objetivo es suavizar las listas de imágenes

En la siguiente captura de pantalla puede ver como usan Glide para traer los thumbnails de los posts:



- Android-state

Usada para guardar o traer el estado de una instancia. A continuación se ve como Reddit hace su propia implementación de los métodos save y restore usando dicha librería.


```

1 package com.reddit.frontpage.ui;
2
3 import android.os.Bundle;
4 import com.evernote.android.state.Bundler;
5 import com.evernote.android.state.InjectionHelper;
6 import com.evernote.android.state.Injector.Object;
7 import java.util.HashMap;
8
9 public class BaseScreen$$StateSaver<T extends BaseScreen> extends Object<T> {
10     private static final HashMap<String, Bundler<?>> BUNDLERS = new HashMap<>();
11     private static final InjectionHelper HELPER = new InjectionHelper( str: "com.red
12
13     public void save(T t, Bundle bundle) {
14         HELPER.putBoolean(bundle, str: "suppressScreenViewEvent", t.suppressScreenV
15         HELPER.putString(bundle, str: "eventRequestId", t.eventRequestId);
16     }
17
18     public void restore(T t, Bundle bundle) {
19         t.suppressScreenViewEvent = HELPER.getBoolean(bundle, str: "suppressScreenV
20         t.eventRequestId = HELPER.getString(bundle, str: "eventRequestId");
21     }
22 }

```

- [Store](#)

Para el manejo de dos capas de almacenamiento (memoria y disco). Facilita el manejo de peticiones a memoria o a la red.

A continuación un ejemplo de su uso trayendo los links de post populares:

```

final class RedditLinkRepository$popularLinkStore$2 extends Lambda implements Function0<Store<Listing<? extends Link>
    final /* synthetic */ RedditLinkRepository f19415a;

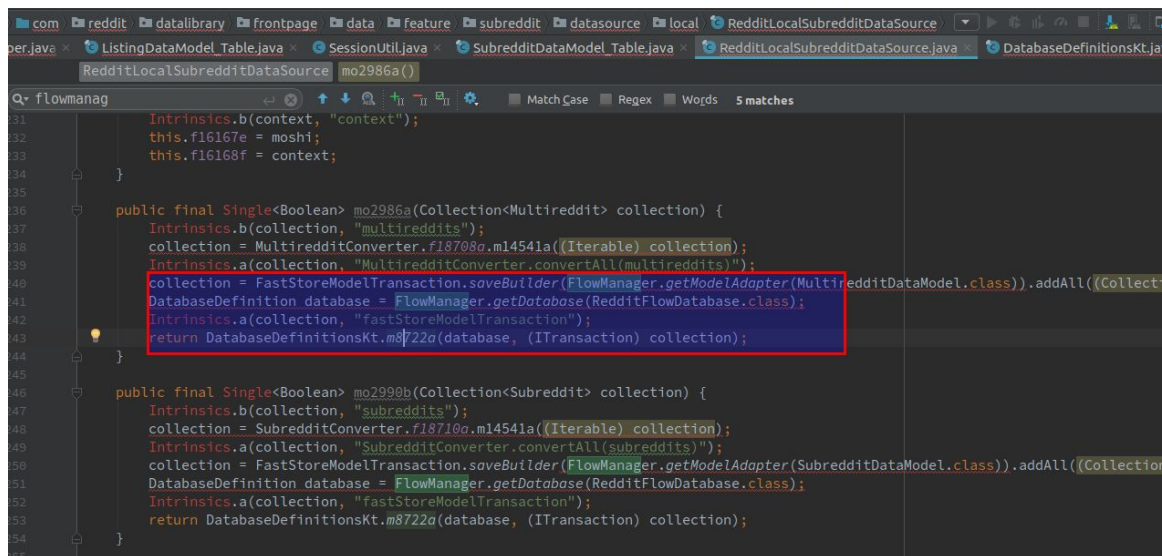
    RedditLinkRepository$popularLinkStore$2(ReditLinkRepository redditLinkRepository) {
        this.f19415a = redditLinkRepository;
        super(0);
    }

    public final /* synthetic */ Object invoke() {
        RealStoreBuilder a = StoreBuilder.m8709a();
        a.f10249b = new Fetcher<Listing<? extends Link>, RequestKey>() {
            public final /* synthetic */ Single.mo2915a(Object obj) {
                RequestKey requestKey = (RequestKey) obj;
                Intrinsics.b(requestKey, "<name for destructuring parameter 0>");
                SortType sortType = requestKey.f10706a;
                SortTimeFrame sortTimeFrame = requestKey.f10707b;
                String str = requestKey.f10708c;
                String str2 = requestKey.f10709d;
                ListingViewMode listingViewMode = requestKey.f10710e;
                RemoteLinkDatasource d = this.f19415a.f16097h;
                if (sortType == null) {
                    Intrinsics.a();
                }
                return DefaultImpls.getPopularLinks$default(d, sortType, sortTimeFrame, str, str2, false, listingV
            }
        };
    }
}

```

- [DBFlow](#)

En este fragmento de código se puede ver como se usa FlowManager para obtener una base de datos SQLite:



```
com reddit datablibrary frontpage data feature subreddit datasource local RedditLocalSubredditDataSource
ListingDataModel_Table.java SessionUtil.java SubredditDataModel_Table.java RedditLocalSubredditDataSource.java DatabaseDefinitionsKt.java

flowmanag

21 Intrinsic.b(context, "context");
22 this.fl6167e = moshi;
23 this.fl6168f = context;
24 }
25
26 public final Single<Boolean> mo2986a(Collection<Multireddit> collection) {
27     Intrinsic.b(collection, "multireddits");
28     collection = MultiredditConverter.fl8708a.m14541a((Iterable) collection);
29     Intrinsic.a(collection, "MultiredditConverter.convertAll(multireddits)");
30     collection = FastStoreModelTransaction.saveBuilder(FlowManager.getModelAdapter(MultiredditDataModel.class)).addAll((Collection) collection);
31     DatabaseDefinition database = FlowManager.getDatabase(RedditFlowDatabase.class);
32     Intrinsic.a(collection, "fastStoreModelTransaction");
33     return DatabaseDefinitionsKt.m8722a(database, (ITransaction) collection);
34 }
35
36 public final Single<Boolean> mo2990b(Collection<Subreddit> collection) {
37     Intrinsic.b(collection, "subreddits");
38     collection = SubredditConverter.fl8710a.m14541a((Iterable) collection);
39     Intrinsic.a(collection, "SubredditConverter.convertAll(subreddits)");
40     collection = FastStoreModelTransaction.saveBuilder(FlowManager.getModelAdapter(SubredditDataModel.class)).addAll((Collection) collection);
41     DatabaseDefinition database = FlowManager.getDatabase(RedditFlowDatabase.class);
42     Intrinsic.a(collection, "fastStoreModelTransaction");
43     return DatabaseDefinitionsKt.m8722a(database, (ITransaction) collection);
44 }
45 }
```

- [Sendbird](#) , usada para el servicio de chat integrado
- [Moshi](#), usada para interpretar archivos json
- [DeepLinkDispatch](#), API basada en anotaciones para el manejo de deep links (URLs que llevan al usuario a contenidos específicos de la app)
- [Volley](#), facilita el manejo de peticiones de red, haciendo buen uso de caché y tareas asíncronas.

Otras librerías usadas fueron:

- [Android Priority Job Queue \(Job Manager\)](#)
- [Crashlytics](#)
- [Android video cache](#)
- [Subscaleview](#)
- [Stetho](#)
- [Firebase job dispatcher](#)
- [Instabug](#)
- [Bridge](#)
- [Thrifty](#)
- [MP4Parser](#)

Buenas prácticas

Uso de String builder y apply()

En la clase ChatSharedPreferencesUtil del paquete reddit/social/util vemos como es usado el String builder para obtener un valor de shared preferences. Después al guardar un valor en shared preferences se utiliza apply() lo cual hace que la transacción se realice de manera síncrona evitando sobrecargar el hilo principal.

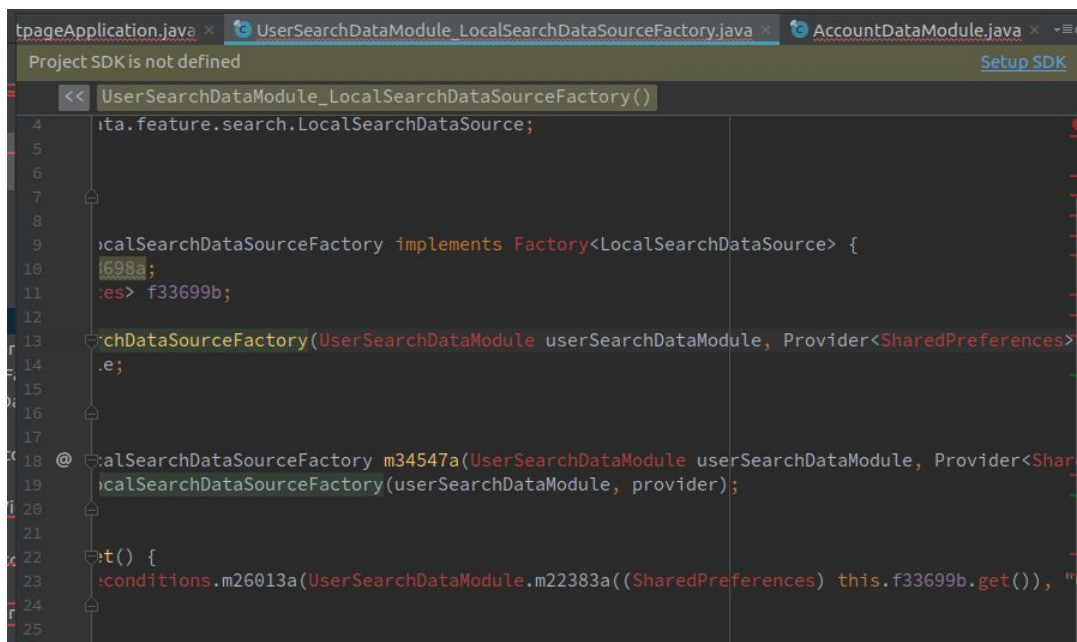
```

public static boolean m24746a(Context context, String str) {
    Account a = AccountUtil.a(context);
    StringBuilder stringBuilder = new StringBuilder(str);
    stringBuilder.append(EventsFileManager.ROLL_OVER_FILE_NAME_SEPARATOR);
    stringBuilder.append(a.name);
    str = stringBuilder.toString();
    boolean z = context.getSharedPreferences("sendBird", 0).getBoolean(str, true);
    if (z) {
        context.getSharedPreferences("sendBird", 0).edit().putBoolean(str, false).apply();
    }
    return z;
}

```

Singleton, Factory

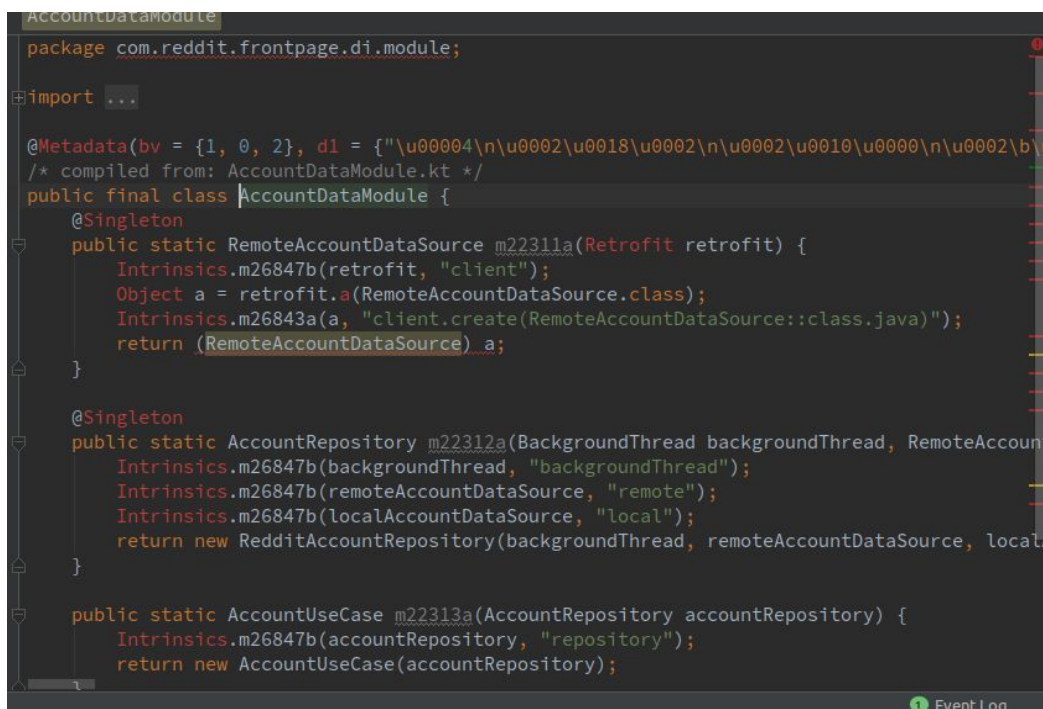
Reddit hace uso de patrones de diseño como lo son Singleton y Factory. En sus clases del paquete reddit/frontpage/di/module se hace uso de una factory por cada module.



```

UserSearchDataSourceModule_LocalSearchDataSourceFactory()
4 ita.feature.search.LocalSearchDataSource;
5
6
7
8
9 LocalSearchDataSourceFactory implements Factory<LocalSearchDataSource> {
10     LocalSearchDataSource f33699b;
11     LocalSearchDataSourceFactory(UserSearchDataSourceModule userSearchDataSourceModule, Provider<SharedPreferences> provider) {
12         f33699b = new LocalSearchDataSource(userSearchDataSourceModule, provider);
13     }
14
15     @Override
16     LocalSearchDataSource m34547a(UserSearchDataSourceModule userSearchDataSourceModule, Provider<SharedPreferences> provider) {
17         return new LocalSearchDataSourceFactory(userSearchDataSourceModule, provider);
18     }
19
20     LocalSearchDataSource m26013a(UserSearchDataSourceModule m22383a((SharedPreferences) this.f33699b.get()), "LocalSearchDataSource") {
21         return new LocalSearchDataSource(m22383a, "LocalSearchDataSource");
22     }
23
24
25

```



```

AccountDataModule
package com.reddit.frontpage.di.module;

import ...

@Metadata(bv = {1, 0, 2}, d1 = {"\u00004\n\u00002\u00018\u0002\n\u00002\u00010\u00000\n\u00002\b"}, d2 = {"com.reddit.frontpage.di.module.AccountDataModule"}, k = 1, mv = {1, 5, 0}, n = 1, p = "AccountDataModule.kt", r = 1, version = 1})
/* compiled from: AccountDataModule.kt */
public final class AccountDataModule {
    @Singleton
    public static RemoteAccountDataSource m22311a(Retrofit retrofit) {
        Intrinsics.m26847b(retrofit, "client");
        Object a = retrofit.a(RemoteAccountDataSource.class);
        Intrinsics.m26843a(a, "client.create(RemoteAccountDataSource::class.java)");
        return (RemoteAccountDataSource) a;
    }

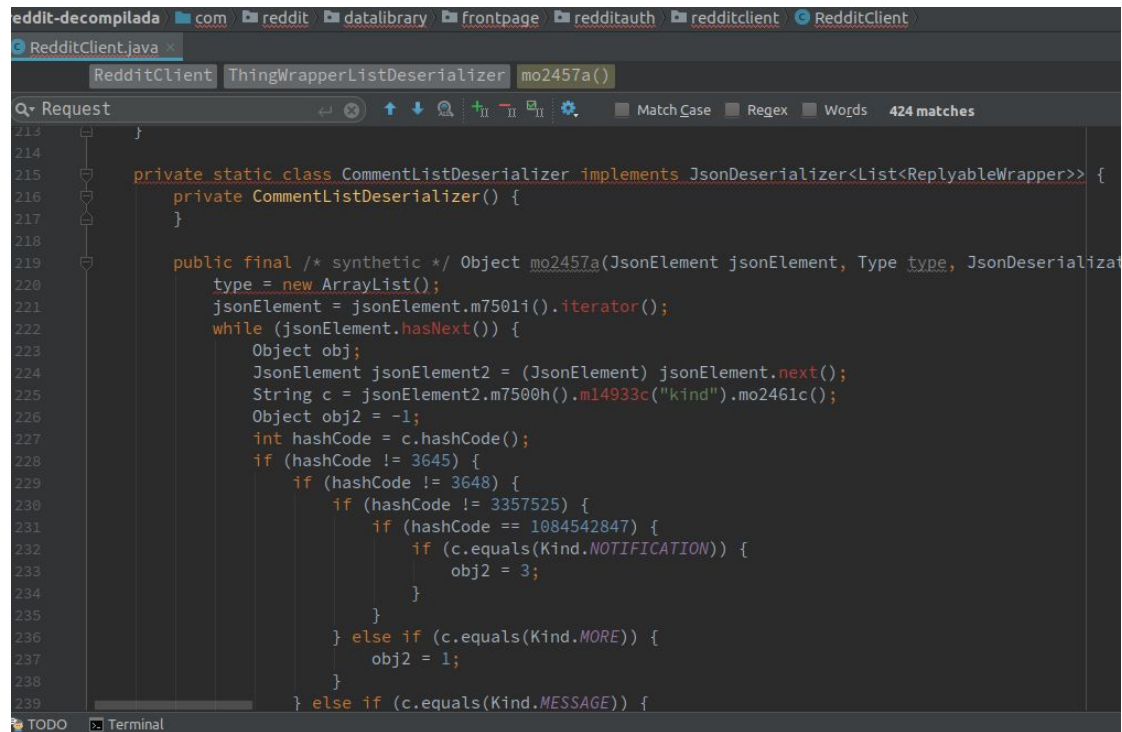
    @Singleton
    public static AccountRepository m22312a(BackgroundThread backgroundThread, RemoteAccountDataSource remoteAccountDataSource, LocalAccountDataSource localAccountDataSource) {
        Intrinsics.m26847b(backgroundThread, "backgroundThread");
        Intrinsics.m26847b(remoteAccountDataSource, "remote");
        Intrinsics.m26847b(localAccountDataSource, "local");
        return new RedditAccountRepository(backgroundThread, remoteAccountDataSource, localAccountDataSource);
    }

    public static AccountUseCase m22313a(AccountRepository accountRepository) {
        Intrinsics.m26847b(accountRepository, "repository");
        return new AccountUseCase(accountRepository);
    }
}

```

Malas prácticas

Se encontró que se están creando objetos dentro de la clase `RedditClient`, al momento de deserializar un elemento json con sus comentarios:



```
RedditClient.java x
RedditClient ThingWrapperListDeserializer mo2457a()

Q: Request
213 }
214
215 private static class CommentListDeserializer implements JsonSerializer<List<ReplyableWrapper>> {
216     private CommentListDeserializer() {
217     }
218
219     public final /* synthetic */ Object mo2457a(JsonElement jsonElement, Type type, JsonSerializer<List<ReplyableWrapper>> serializer) {
220         type = new ArrayList();
221         jsonElement = jsonElement.m7501i().iterator();
222         while (jsonElement.hasNext()) {
223             Object obj;
224             JsonElement jsonElement2 = (JsonElement) jsonElement.next();
225             String c = jsonElement2.m7500h().m14933c("kind").mo2461c();
226             Object obj2 = -1;
227             int hashCode = c.hashCode();
228             if (hashCode != 3645) {
229                 if (hashCode != 3648) {
230                     if (hashCode != 3357525) {
231                         if (hashCode == 1084542847) {
232                             if (c.equals(Kind.NOTIFICATION)) {
233                                 obj2 = 3;
234                             }
235                         } else if (c.equals(Kind.MORE)) {
236                             obj2 = 1;
237                         }
238                     } else if (c.equals(Kind.MESSAGE)) {
239                         obj2 = 2;
240                     }
241                 }
242             }
243             obj = serializer.serialize(jsonElement2, type);
244             type.add(obj);
245         }
246         return type;
247     }
248 }
```

Estrategias de almacenamiento

Shared preferences

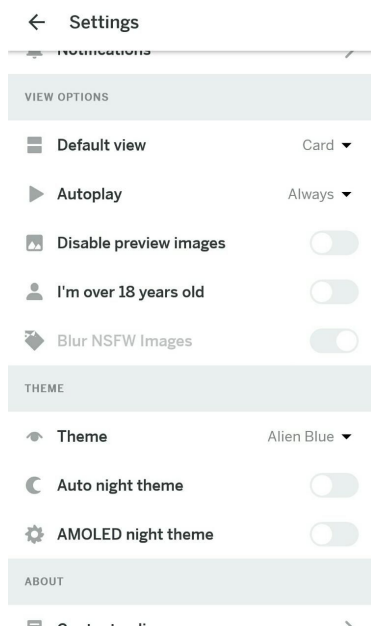
Es usado para guardar preferencias básicas de la aplicación. En este caso, se están guardando cosas como modo nocturno activado o no, si se desea ver posts en cards o de manera compacta. A continuación el fragmento de código de la clase `UpgradeUtil.java` en el paquete `reddit/frontpage/util`

```

}
if (i <= 20500) {
    i = FrontpageApplication.f27402a.getSharedPreferences("com.reddit.frontpage.settings", 0);
    FrontpageSettings a2 = FrontpageSettings.a();
    a2.a(i.getBoolean("com.reddit.mobile.disable_preview_images", false));
    a2.a(i.getBoolean("com.reddit.pref.compact_listing_view", false), false);
    a2.b(i.getBoolean("com.reddit.pref.night_mode", false));
    a2.e(i.getBoolean("com.reddit.pref.blur_nsfw", true));
    a2.a(i.getString("com.reddit.pref.autoplay", Util.m24027f((int) C1761R.string.option_value_
    a2.b(i.getString("com.reddit.pref.last_push_token", null));
    a2.c(i.getString("com.reddit.pref.last_push_token_user", null));
    a2.d(i.getString("com.reddit.pref.base_uri", Util.m24027f((int) C1761R.string.base_uri_defa
    a2.e(i.getString("com.reddit.pref.gateway_uri", Util.m24027f((int) C1761R.string.gateway_ur
    a2.g(i.getBoolean("com.reddit.pref.dummy_upcoming_event", false));
    InternalSettings.a().a(i.getInt("com.reddit.pref.old_version", RedditJobManager.f10810d));
    i.edit().clear().apply();
    if (SessionManager.b().c.isAnonymous() == 0) {
        i = FrontpageApplication.f27402a.getSharedPreferences("com.reddit.frontpage.internal_se
        SharedPreferences sharedPreferences = FrontpageApplication.f27402a.getSharedPreferences
        sharedPreferences.edit().putLong( s: "com.reddit.frontpage.last_upvote_timestamp", i.get
        sharedPreferences.edit().putBoolean( s: "com.reddit.frontpage.rage_shake_unlocked", i.ge
    }
}

```

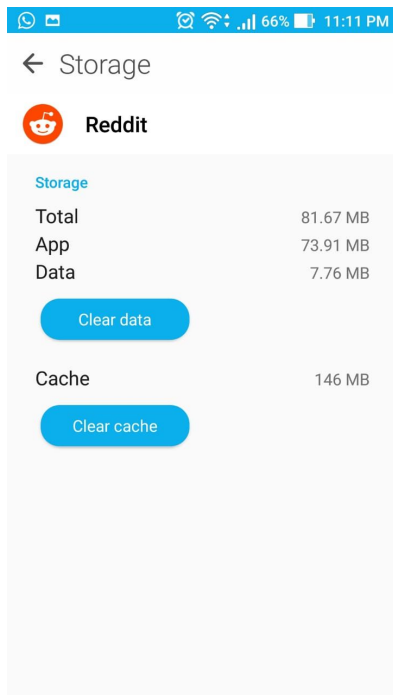
Estas preferencias se pueden ver en los ajustes de la aplicación:



Manejo de caché

La aplicación presenta un gran consumo de almacenamiento y guarda mucha información en disco. Esto fue comprobado cuando al reiniciar el celular en modo sin conexión, todavía se podían visualizar los post guardados la última vez que esta se abrió. Esto se debe a que la librería que usan, Glide tiene opta por redimensionar las imágenes que necesita la app y guardar una copia de esta por cada dimensión. Esto ocupa más espacio, pero hace más eficiente el cargue de la imagen en la aplicación debido a que no tiene que redimensionarla cada vez que es cargada. Glide implementa LRUCache como estructura de caching según

su documentación <https://bumptech.github.io/glide/doc/caching.html> . A continuación, se comprueba que se tienen 146mb almacenados en caché pero solo se tenía 3 días de



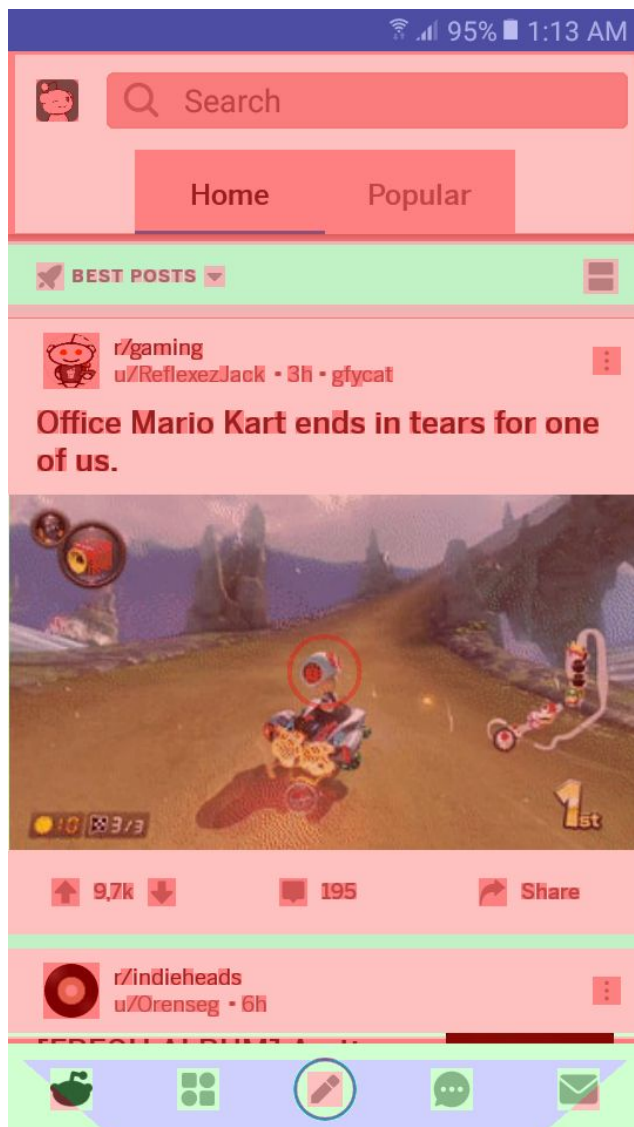
instalada con un uso ocasional de aproximadamente 20 minutos por día, navegando por post populares y reddit's suscritos. Entre estos la mayoría son imágenes y videos cortos.

UI

Overdraw

El overdraw ocurre cuando sistema repinta un píxel en la pantalla varias veces en un solo cuadro de renderizado. Tendiendo en cuenta que se manifiesta como un problema de rendimiento al perderse tiempo de la GPU para repintar los píxeles, se activo la opcion de desarrollador **Debug GPU overdraw** para inspeccionar la aplicación.

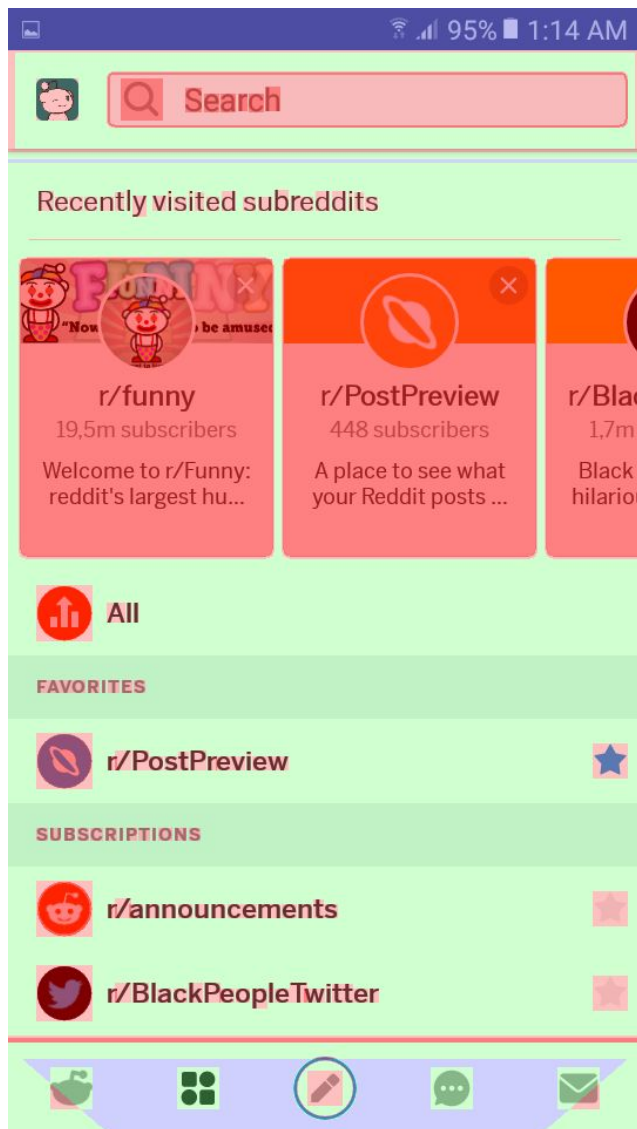
Hallazgo 1: en la vista a cargo de presentar los post al usuario, se puede observar repintado en los pixeles ubicados en el área del plain text de búsqueda. También en los botones a cargo de cambiar los post mostrados en el fragmento entre Home y Popular. Por otra parte se observa un repintado de los píxeles en donde hay un carácter y archivo multimedia.



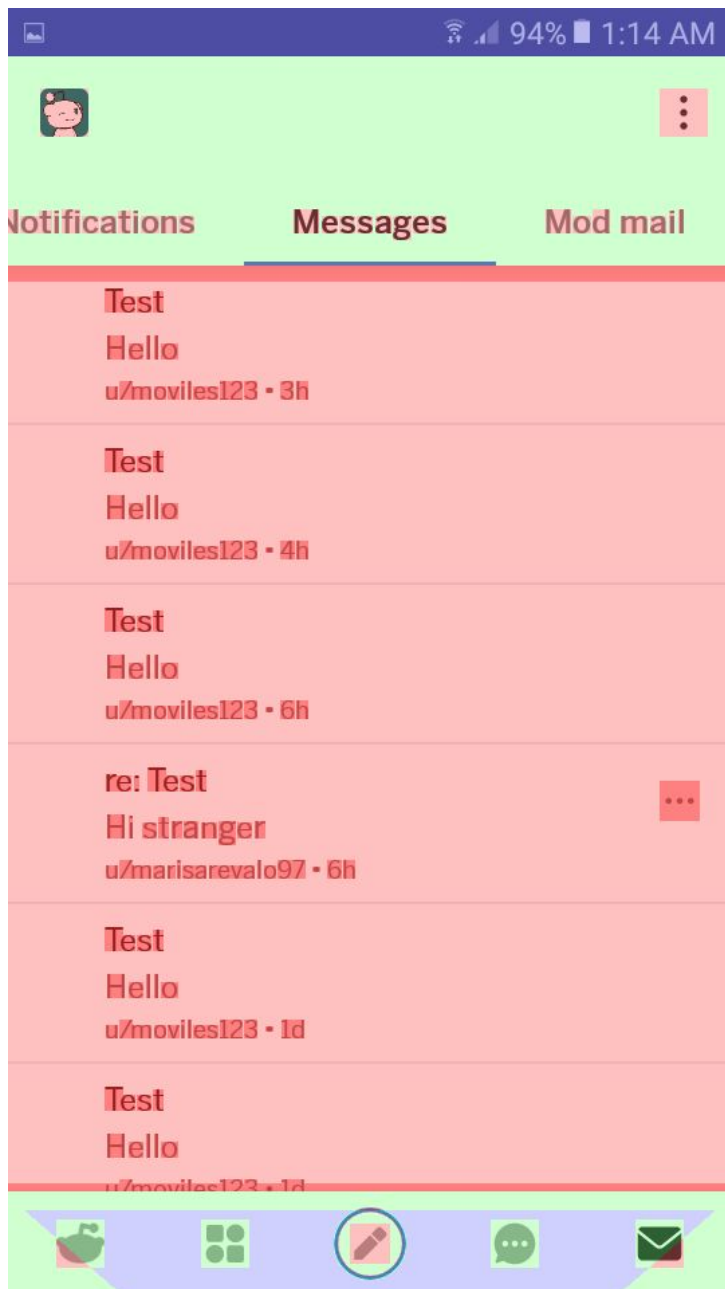
Hallazgo 2: en la vista a cargo de presentar los comentarios de un post, se puede observar que los píxeles ubicados en la barra de like/número de comentarios / share están completamente repintados. También en los botones, en parte porque en esta posición se carga el background y el botón sino también un icono/imagen sobre el botón. Por otra parte se observa un repintado de los píxeles en donde hay un carácter.



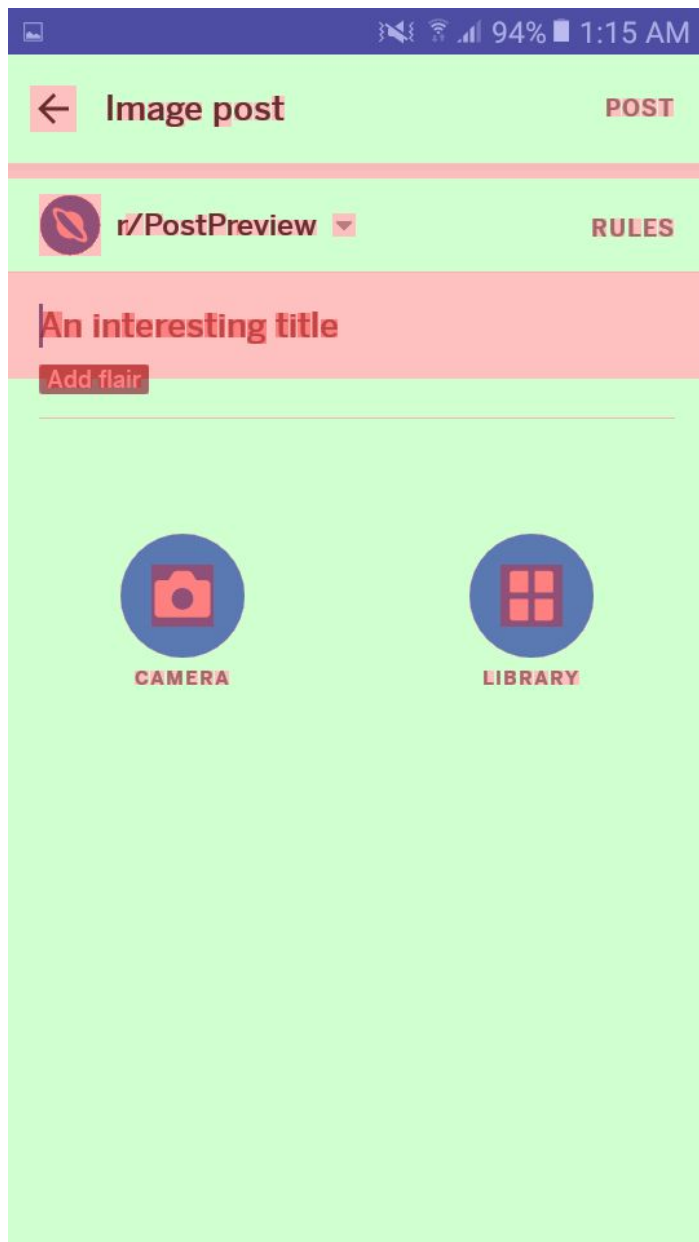
Hallazgo 3: en la vista a cargo de presentar la lista de sub reddit, se puede observar repintado en los pixeles ubicados en el área del plain text de búsqueda. También en los botones, en parte porque en esta posición no solo se carga el background y el botón sino también un icono/imagen sobre el botón. Por otra parte se observa un repintado de los píxeles en donde hay un carácter y las cards.



Hallazgo 4: en la vista a cargo de presentar el buzón de mensajes, se puede observar repintado en los píxeles ubicados en el área del list view. También se observa un repintado de los píxeles en donde hay un carácter.



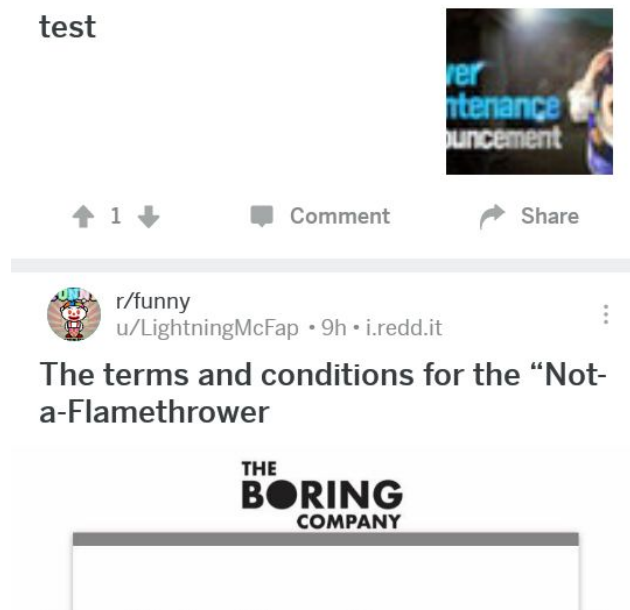
Hallazgo 5: en la vista a cargo de la publicación de un post por parte del usuario, se puede observar repintado en los píxeles ubicados en el área del plain text del título del post. También se observa un repintado de los píxeles en donde hay un carácter y archivo multimedia. Por otra parte, en los píxeles donde hay botones porque en esta posición no solo se carga el background y el botón sino también un icono/imagen sobre el botón.



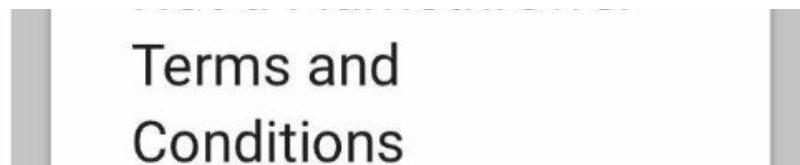
Diseño

Para tratar este tema, se consultó a un estudiante de último semestre de Diseño de la Universidad de los Andes, Sebastian Medina.

De acuerdo a Sebastian: la aplicación presenta un diseño orgánico muy comercial, en general la transición entre tareas es lógica e intuitiva. Sin embargo, carece de contraste, por lo que en dificulta diferenciar los diferentes elementos de una misma vista.



También, opina que los botones de la barra inferior son muchos y pequeños, por lo que dificulta su uso.



Conclusiones

La aplicación Reddit está bien diseñada para su objetivo, visualizar contenido y discutirlo. Opta por hacer más uso de disco con tal de lograr una mejor experiencia de usuario. Sin embargo, se encontraron serias fallas de seguridad en cuanto a su comunicación con otras aplicaciones. Su diseño de interfaz es simple y está orientado a contenido. Sin embargo, carece de mensajes de error explícitos para el usuario. Hace uso de Kotlin y de una gran cantidad de librerías para facilitar procesos de caching, trabajos asíncronos, bases de datos, envío de mensajes, etc.