

Reporte Reddit

Introducción	2
Desempeño	2
Conectividad eventual	12
Seguridad	14
Análisis del código	16
Librerías encontradas	17
Buenas prácticas	20
Malas prácticas	21
Estrategias de almacenamiento	22
Shared preferences	22
Manejo de caché	23
UI-Usabilidad	23
Diseño	25

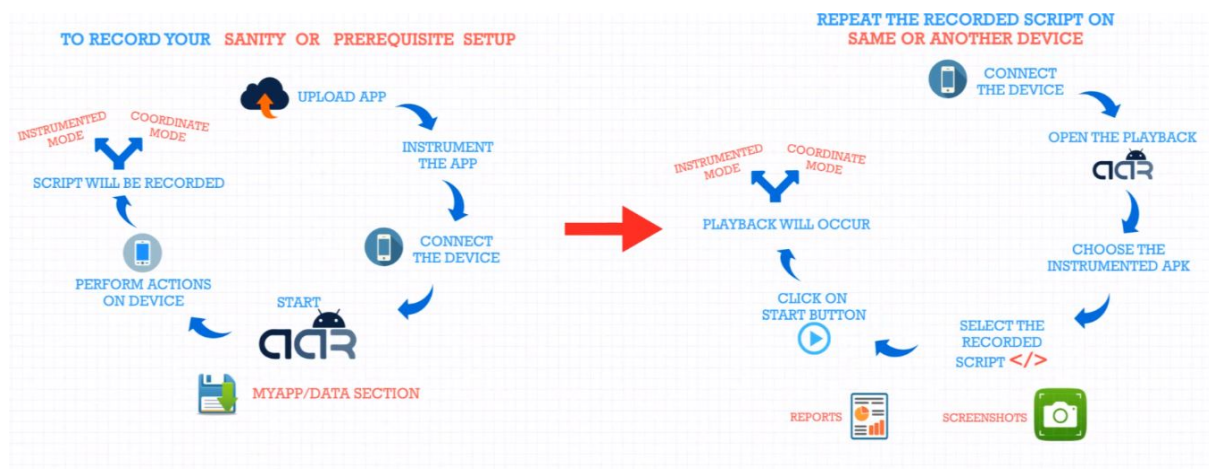
Introducción

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.

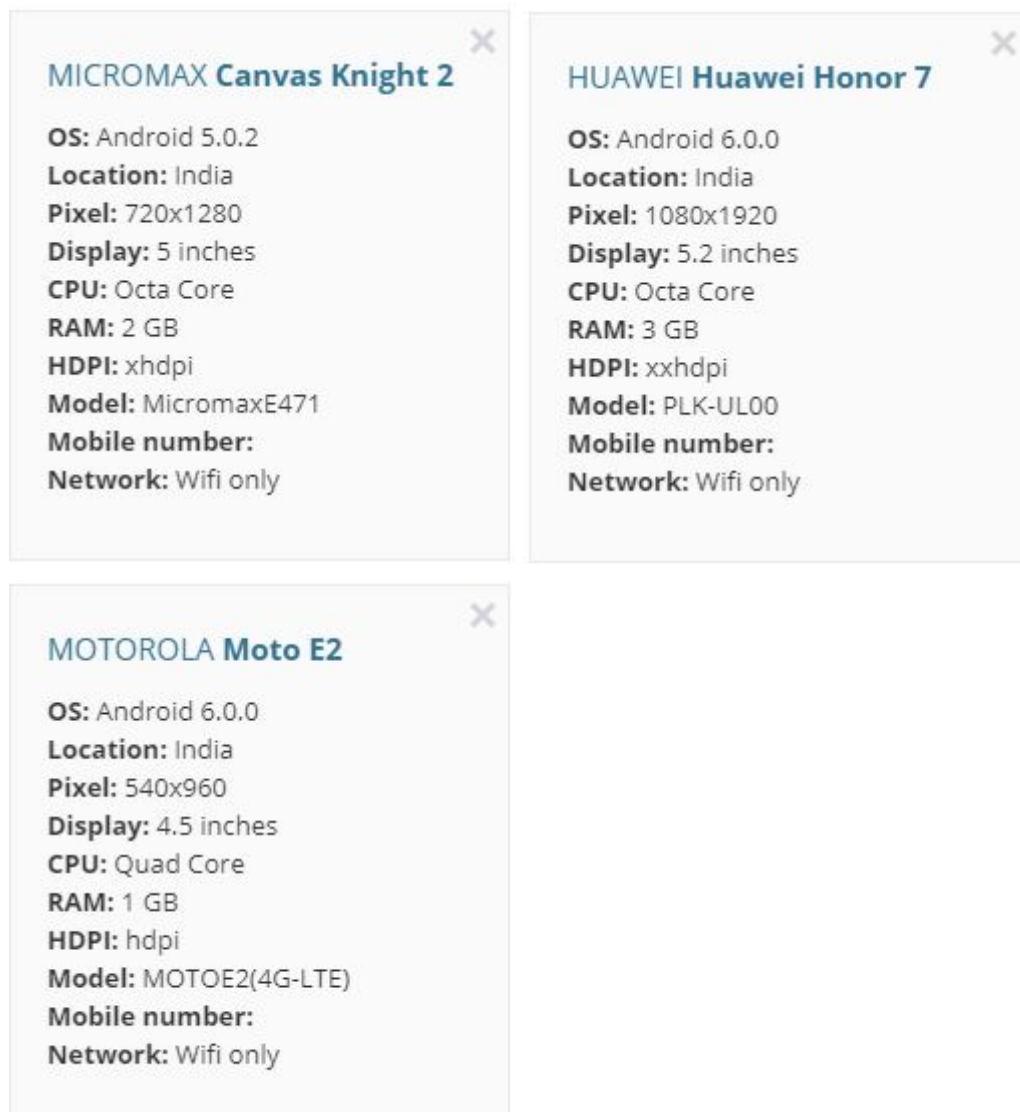
Desempeño

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:

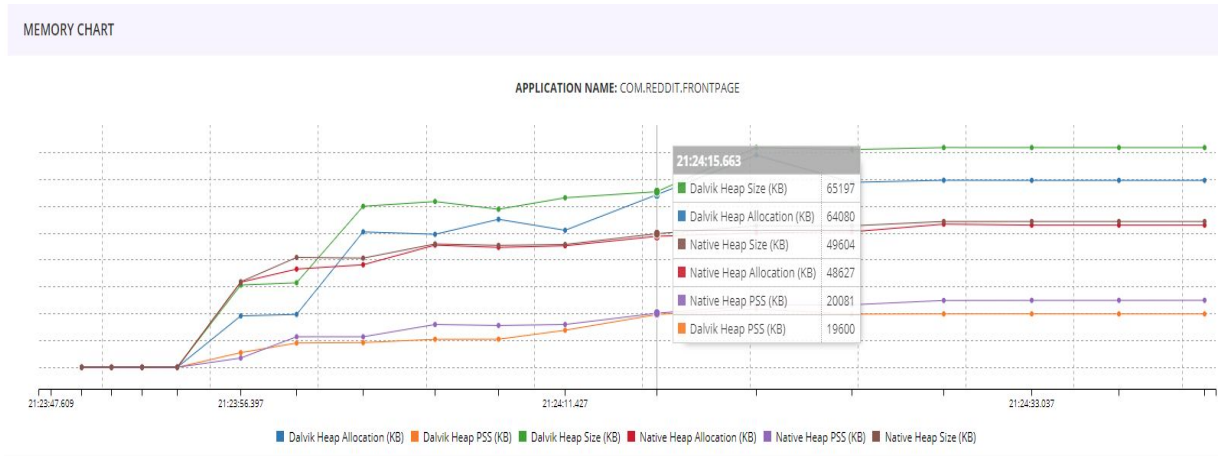
Flujo Uno: El usuario no hace login en la aplicación. Se dirige a la sección Public donde se muestran los post calientes (publicación reciente y número de comentarios y likes). El usuario selecciona un post y hace swipe para leer los comentarios de la aplicación

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

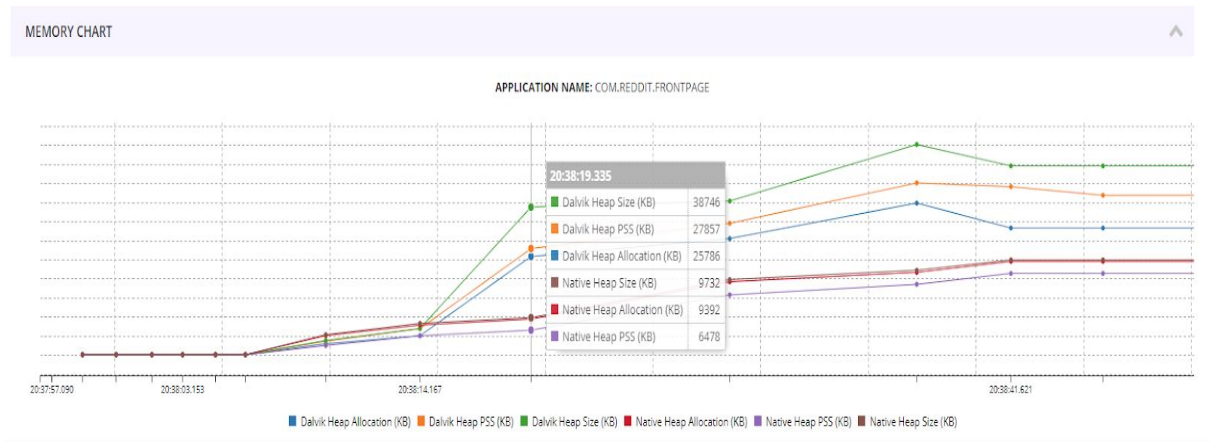
Por otra parte, se realizó una captura en video de la simulación automatizada de este flujo en el dispositivo Moto E2. este video se encuentra en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_1/Flujo1_VIDEO.mp4](#) del repositorio

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

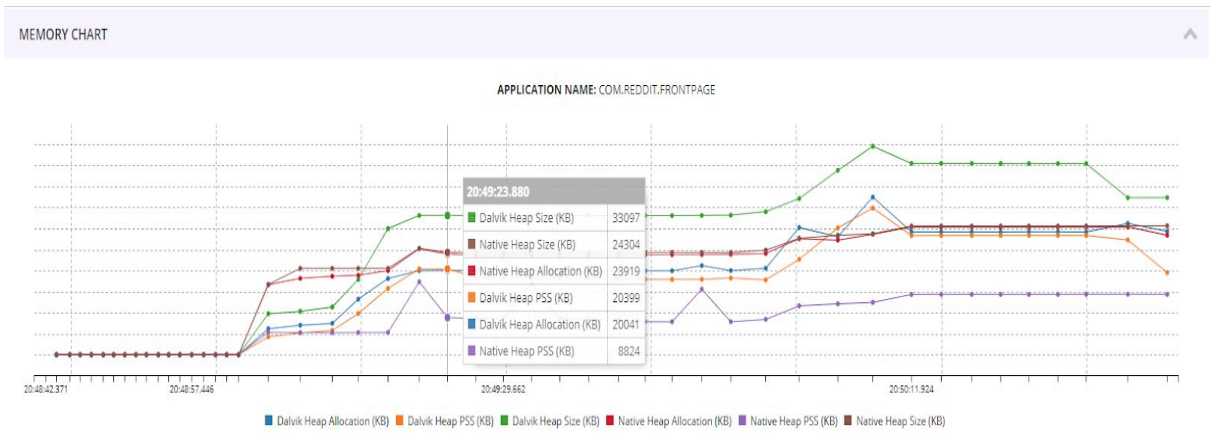
Memoria:



HUAWEI HONOR 7

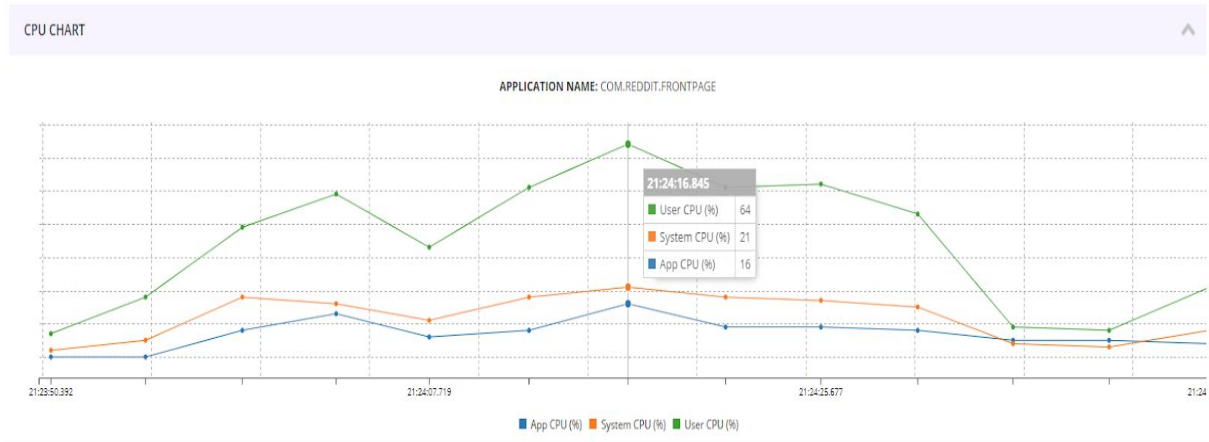


MICROMAX KNIGHT 2

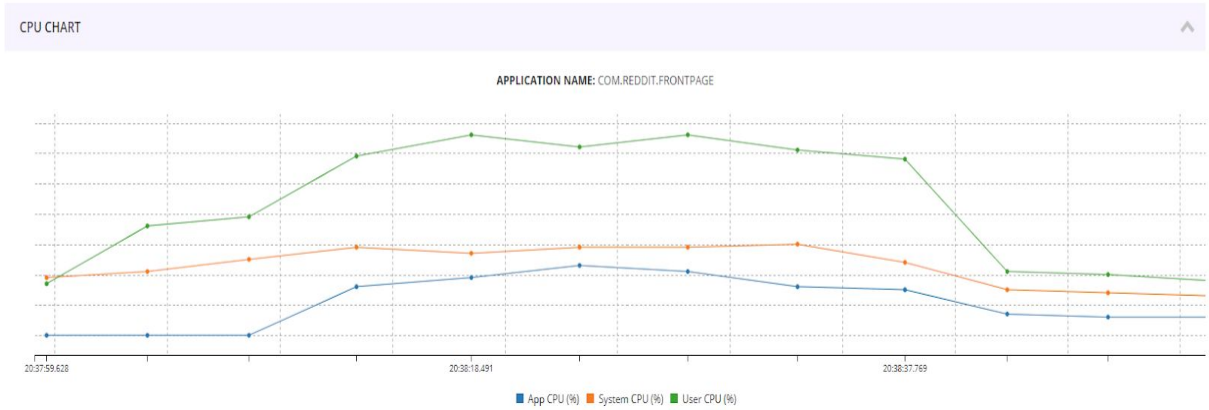


MOTO E2

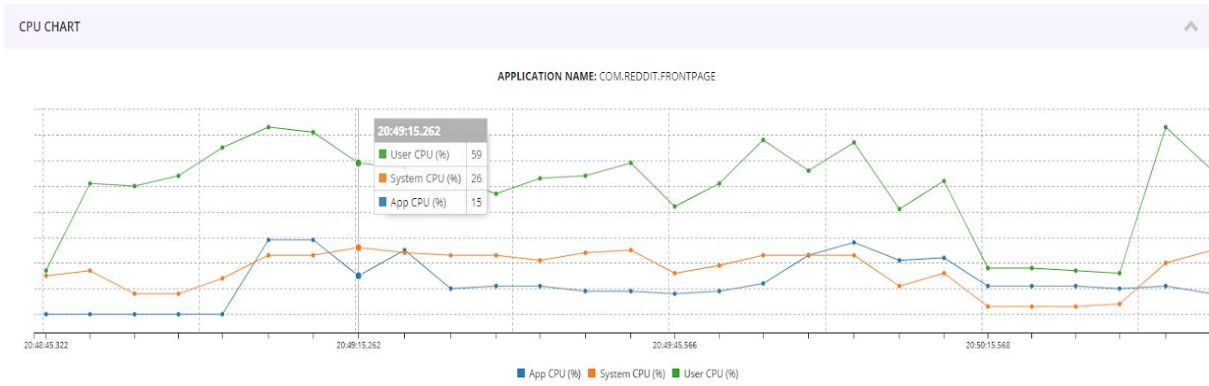
CPU:



HUAWEI HONOR 7

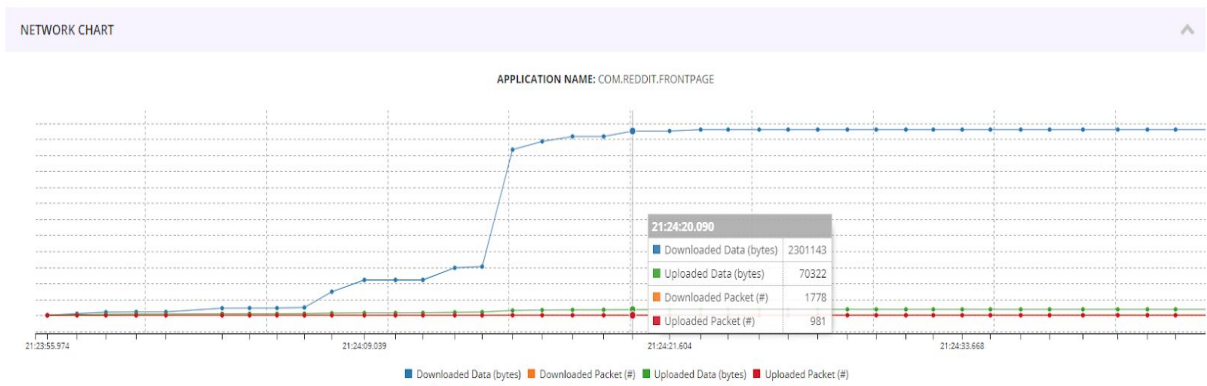


MICROMAX KNIGHT 2

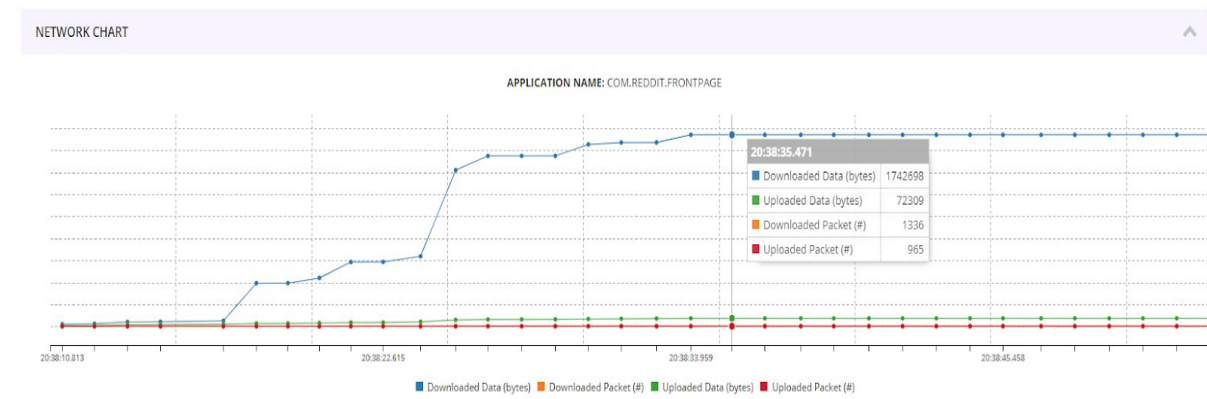


MOTO E2

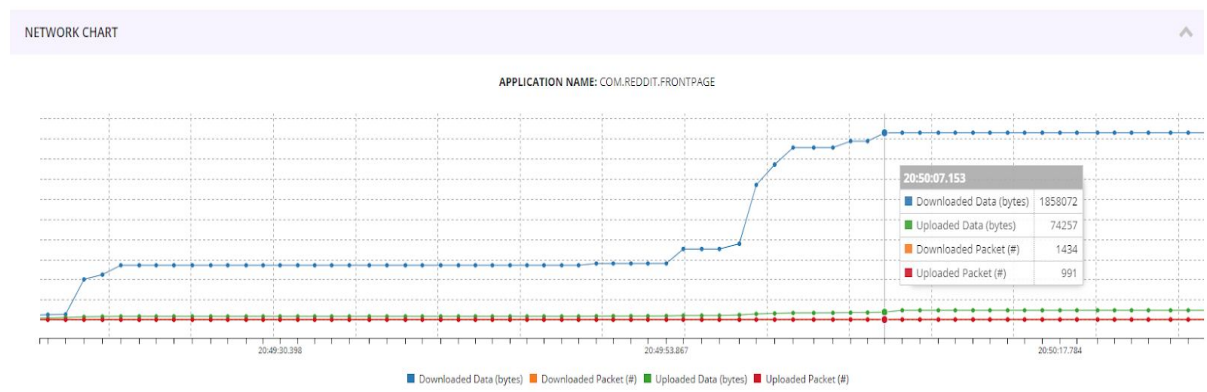
Red:



HUAWEI HONOR 7



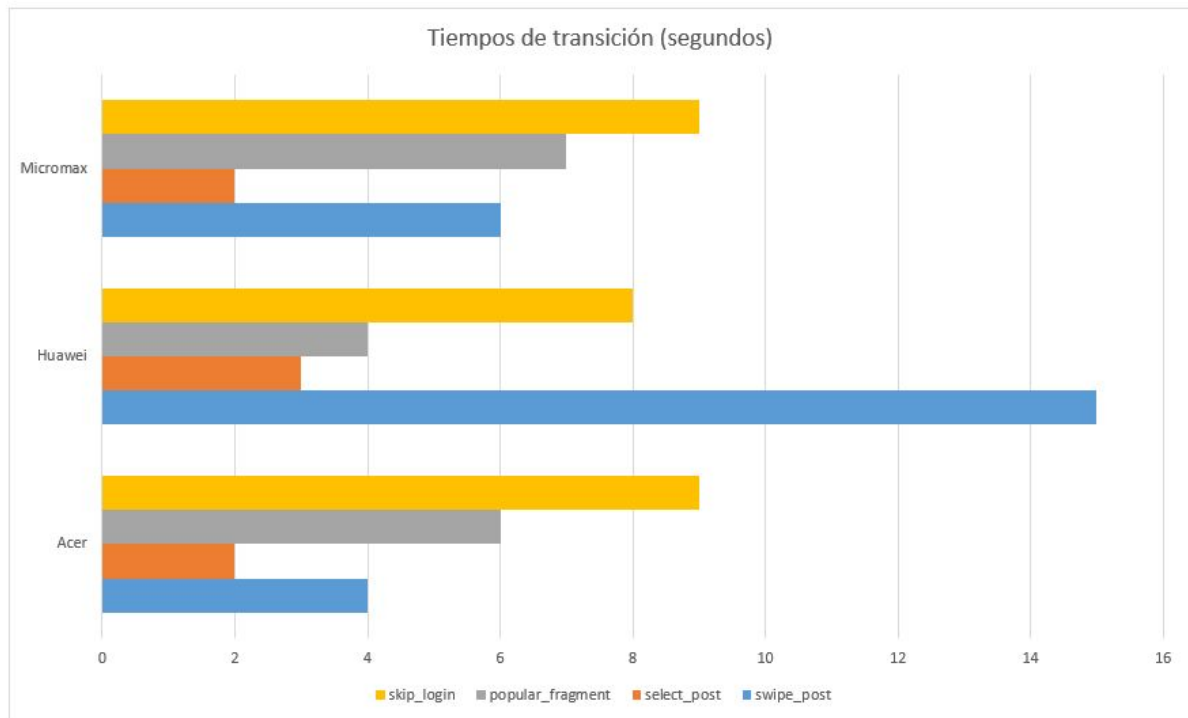
MICROMAX KNIGHT 2



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 de activity a otro



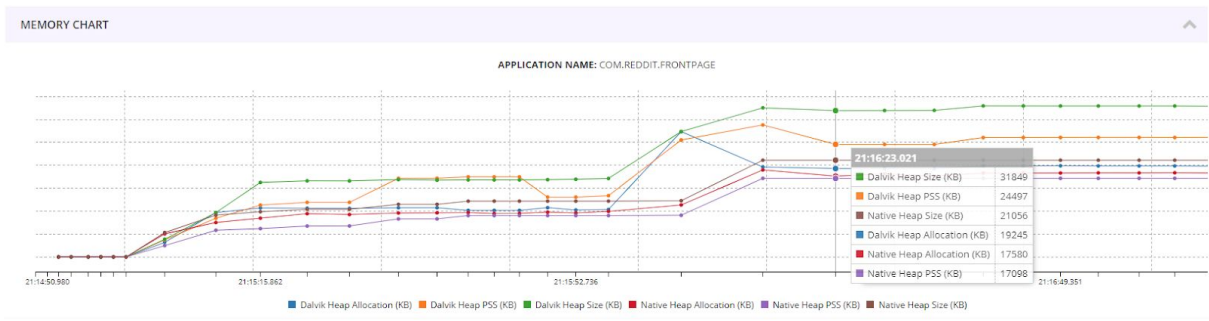
Flujo 2: El usuario hace login. Se dirige a la sección de mensajes, escribe un texto a un destinatario: ingresa destinatario r/<amigo>, ingresa un asunto y el texto del mensaje. Por último, revisa su bandeja de entrada.

Nota: en la dirección **ReporteReddit/VideosFlujosAutomatizados/Flujo_2/** del repositorio se encuentra las capturas de pantalla del flujo paso a paso.

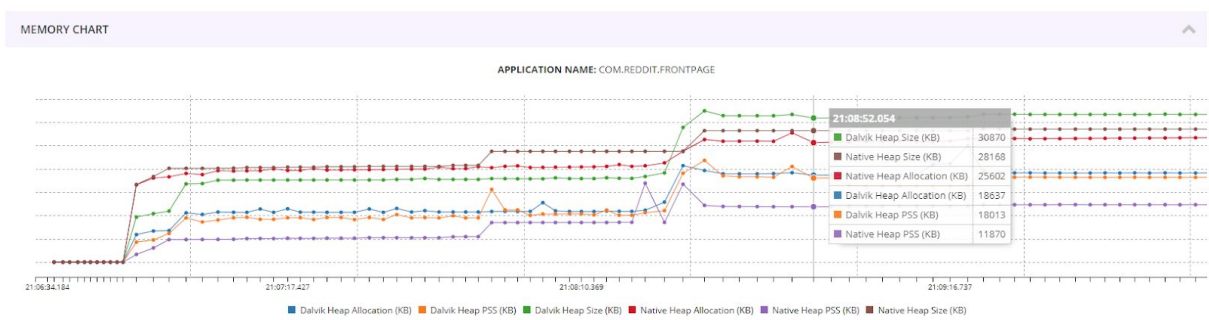
Por otra parte, se realizó una captura en video de la simulación automatizada de este flujo en el dispositivo Moto E2. este video se encuentra en la dirección **ReporteReddit/VideosFlujosAutomatizados/Flujo_2/Flujo2_VIDEO.mp4** del repositorio

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 imagenes dirigirse a la dirección **ReporteReddit/CapturaFlujos/Flujo_2/** del repositorio

Memoria:

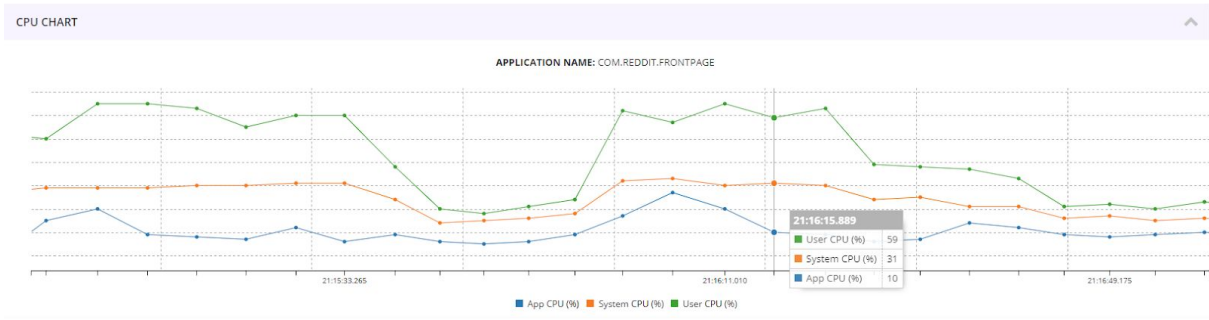


MICROMAX KNIGHT 2

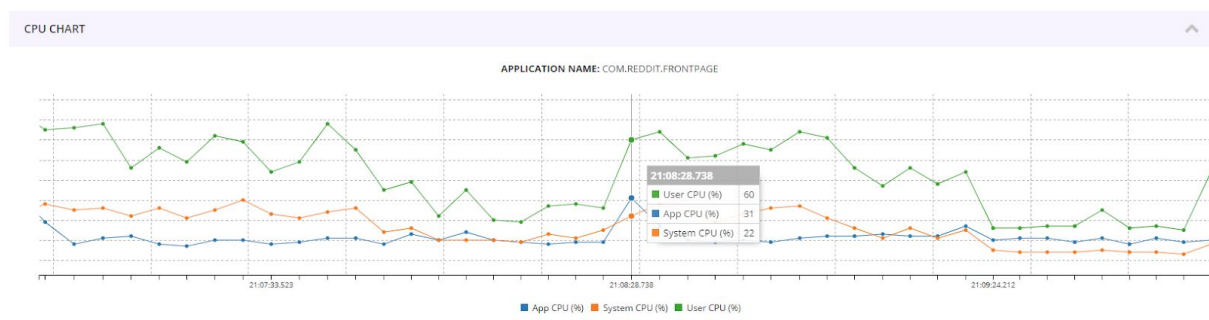


MOTO E2

CPU:

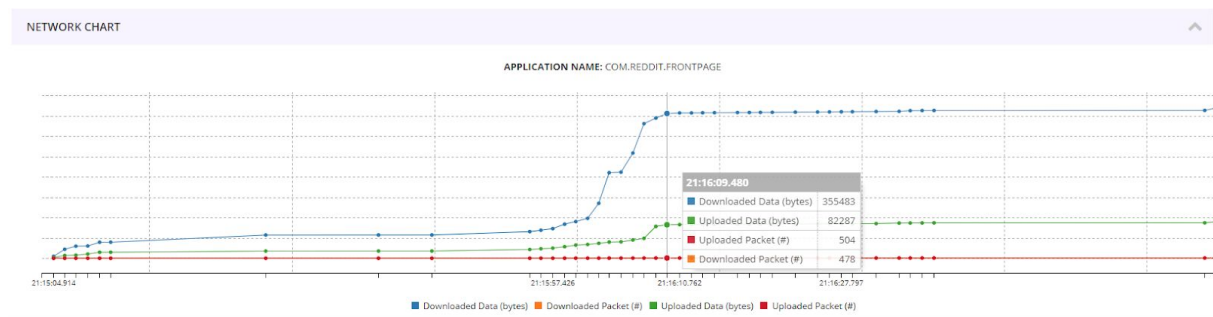


MICROMAX KNIGHT 2

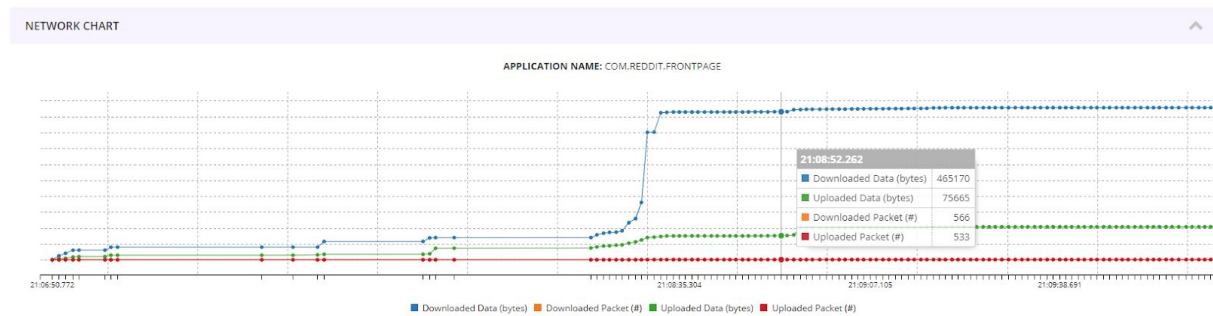


MOTO E2

Red:



MICROMAX KNIGHT 2



MOTO E2

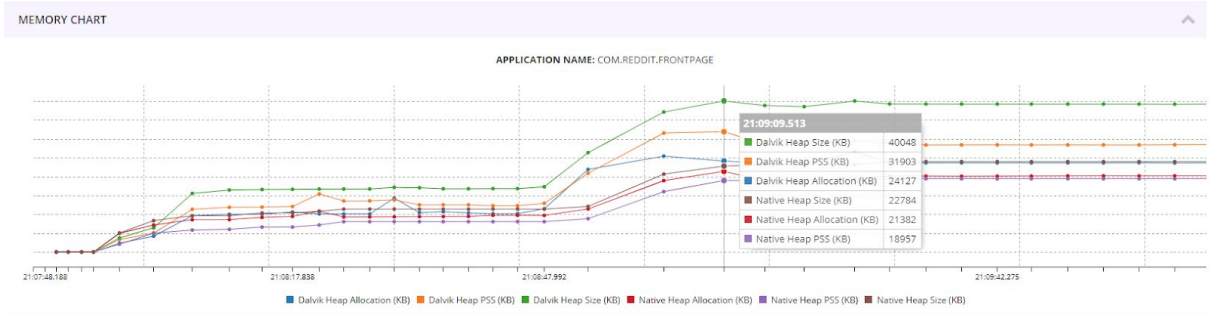
Flujo 3: El usuario hace login. Se dirige a los subreddits a los que está inscrito y selecciona uno de estos foros. Posteriormente, selecciona el botón de publicación de texto e ingresa el nombre del post y su contenido. Finalmente da click en post.

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

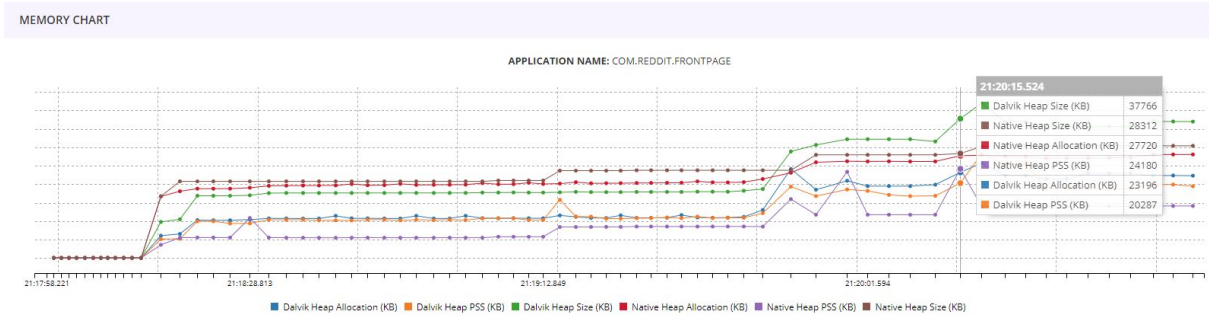
Por otra parte, se realizó una captura en video de la simulación automatizada de este flujo en el dispositivo Moto E2. este video se encuentra en la dirección [ReporteReddit/VideosFlujosAutomatizados/Flujo_3/Flujo3_VIDEO.mp4](#) del repositorio

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 imagenes dirigirse a la dirección [ReporteReddit/CapturaFlujos/Flujo_3/](#) del repositorio

Memoria

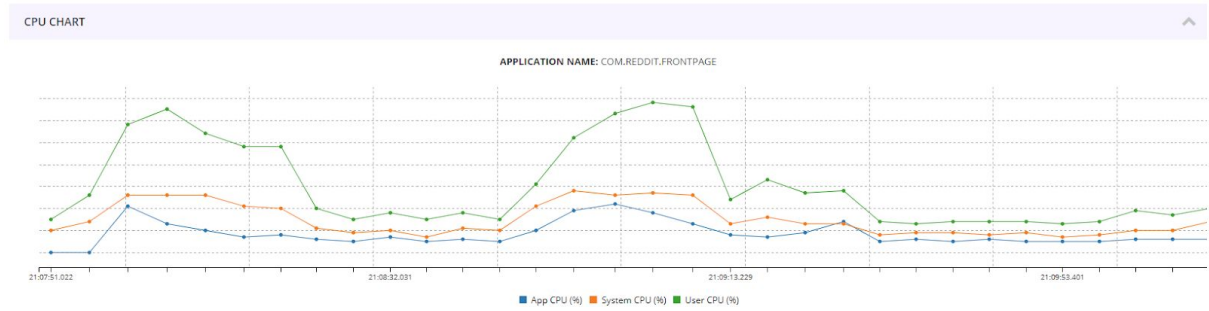


MICROMAX KNIGHT 2

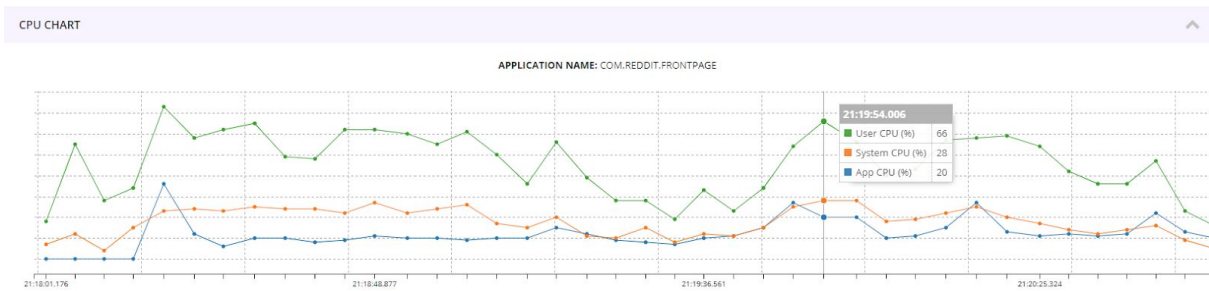


MOTO E2

CPU

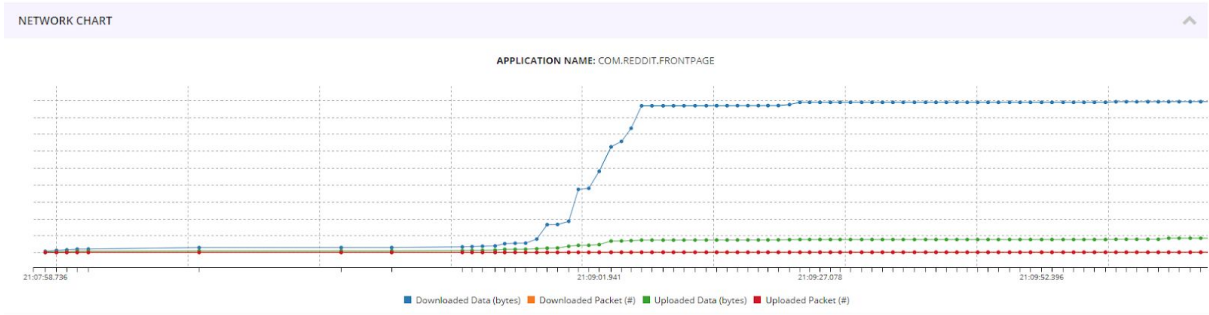


MICROMAX KNIGHT 2

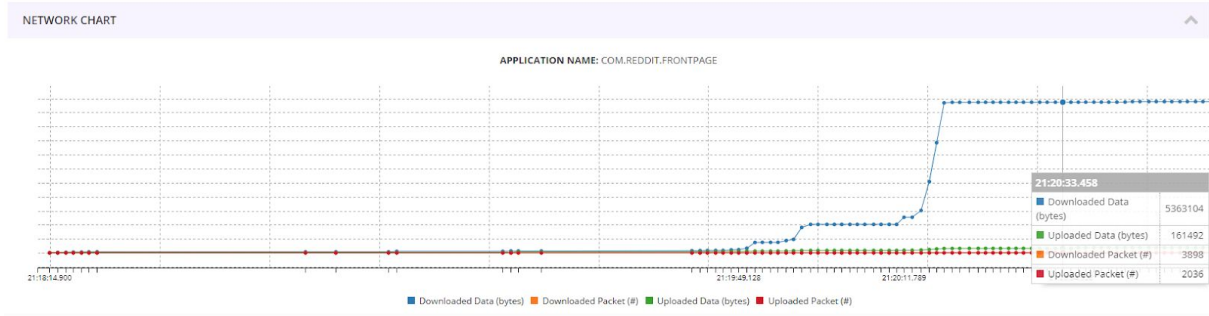


MOTO E2

Red



MICROMAX KNIGHT 2



MOTO E2

Tiempos de instalación y corrido en distintos dispositivos

Device Statuses (2)

☒ Device

Tests

Time Bar

Dev...

☐ LG Google Nexus 5 6.0.1 (...)
Android 6.0.1

100%
1/1

43s

1m 31s

of 2m 14s

Succeeded

☐ LG Google Nexus 5 D820 ...
Android 5.0

100%
1/1

48s

4s

1m 24s

of 2m 16s

Succeeded

Legend

Installing

Launching

Running

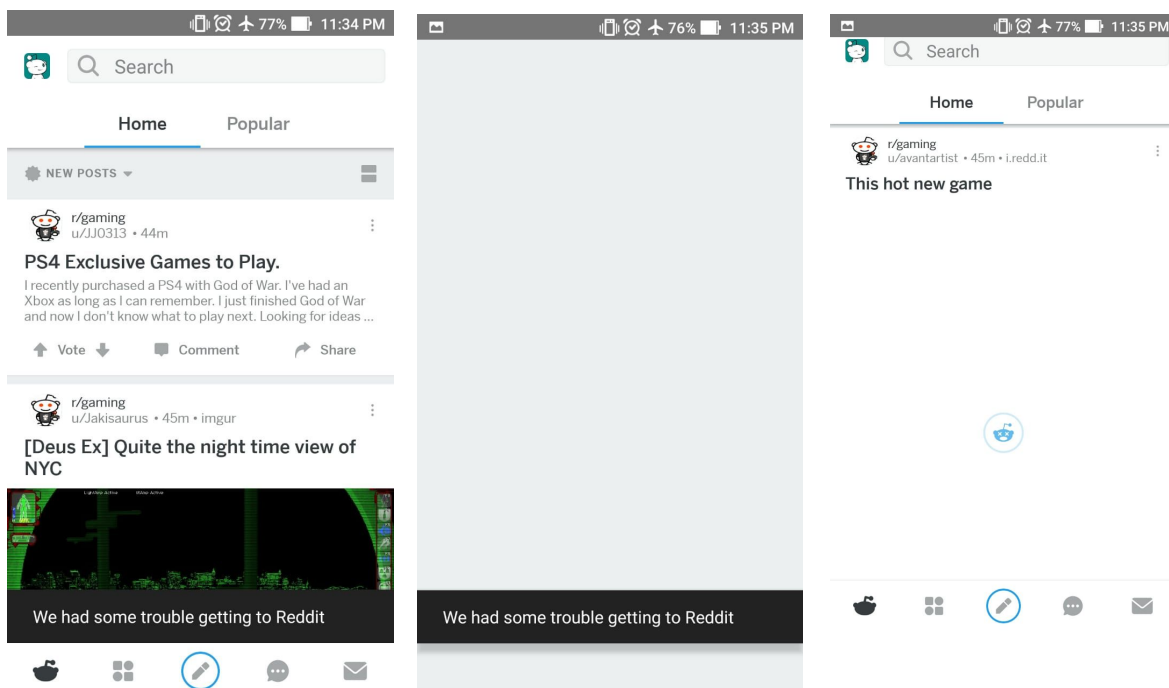
Conectividad eventual

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

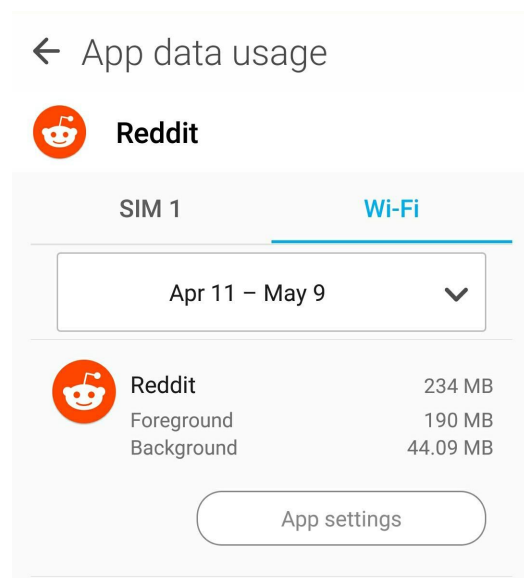
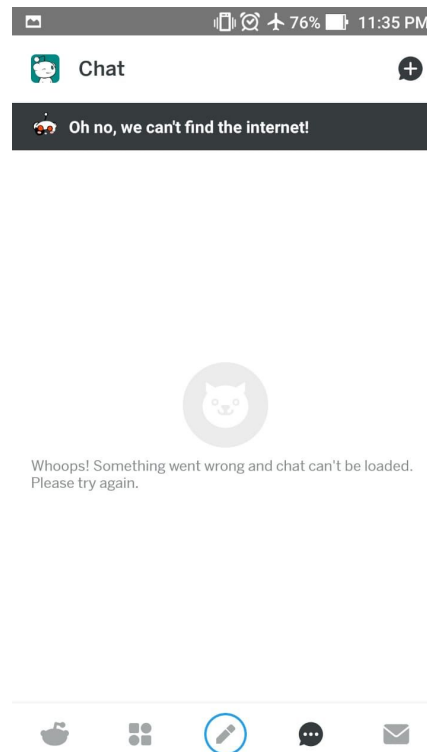
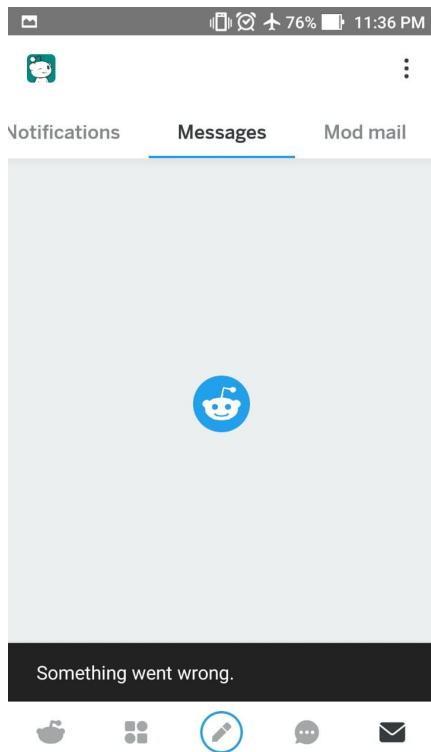
En la primera imagen se tenía wifi activado en la lista de mensajes, sin embargo, se desactivo antes de entrar a un mensaje en particular.

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.

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/WSCliet.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/utils/SimpleMultipartEntity.java okio/Options.java
---	------	---

El uso de md5 tanto en las librerías como en el package de reddit

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

MD5 is a weak hash known to have hash collisions.	high	com/danikula/video/cache/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
---	------	--

También 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.reddithash_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

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:

- [Cookpad](#)
- [Crashlytics](#)
- [Android video cache](#)
- [Subscaleview](#)
- [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.InjectorHelper;
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 InjectorHelper HELPER = new InjectorHelper( str: "com.redc
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 }

```

- [Stetho](#)
- [Firebase job dispatcher](#)
- [Instabug](#)
- [Bridge](#)
- [Thrifty](#)
- [MP4Parser](#)
- [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;

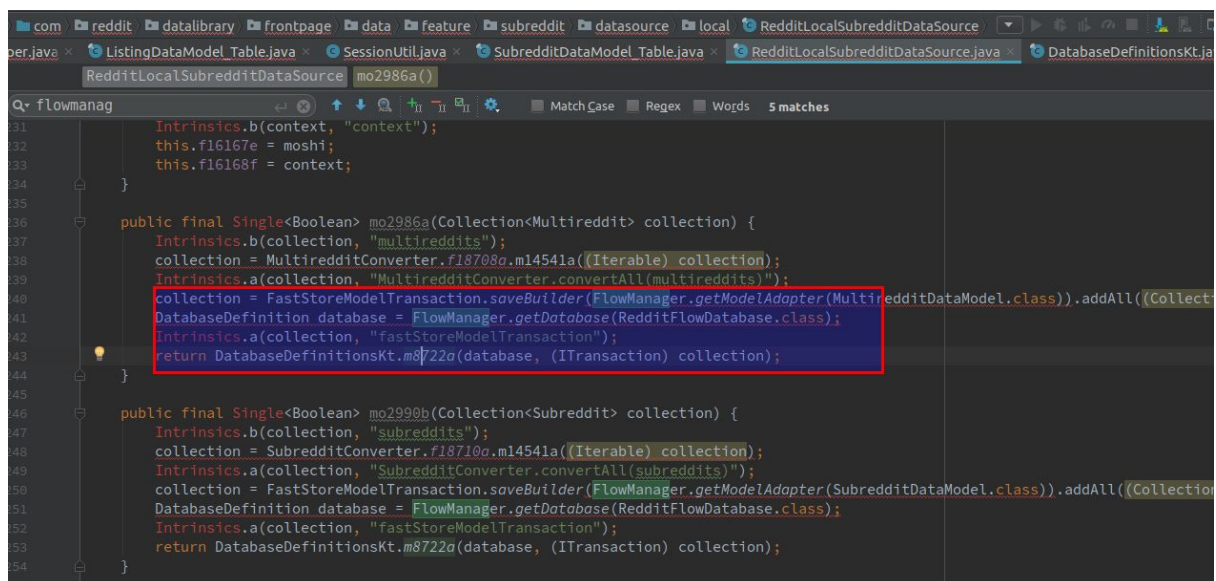
    RedditLinkRepository$popularLinkStore$2(RedditLinkRepository redditLinkRepository) {
        this.f19415a = redditLinkRepository;
        super(0);
    }

    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:



```

    public final Single<Boolean> mo2986a(Collection<Multireddit> collection) {
        Intrinsics.b(collection, "multireddits");
        collection = MultiredditConverter.f18708a.m14541a((Iterable) collection);
        Intrinsics.a(collection, "MultiredditConverter.convertAll(multireddits)");
        collection = FastStoreModelTransaction.saveBuilder(FlowManager.getModelAdapter(MultiredditDataModel.class)).addAll((Collection) collection);
        DatabaseDefinition database = FlowManager.getDatabase(RedditFlowDatabase.class);
        Intrinsics.a(collection, "fastStoreModelTransaction");
        return DatabaseDefinitionsKt.m8722a(database, (ITransaction) collection);
    }

    public final Single<Boolean> mo2990b(Collection<Subreddit> collection) {
        Intrinsics.b(collection, "subreddits");
        collection = SubredditConverter.f18710a.m14541a((Iterable) collection);
        Intrinsics.a(collection, "SubredditConverter.convertAll(subreddits)");
        collection = FastStoreModelTransaction.saveBuilder(FlowManager.getModelAdapter(SubredditDataModel.class)).addAll((Collection) collection);
        DatabaseDefinition database = FlowManager.getDatabase(RedditFlowDatabase.class);
        Intrinsics.a(collection, "fastStoreModelTransaction");
        return DatabaseDefinitionsKt.m8722a(database, (ITransaction) collection);
    }

```

- [Sendbird](#)
- [Picasso](#)
- [Moshi](#)
- [Twitter sdk](#)

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

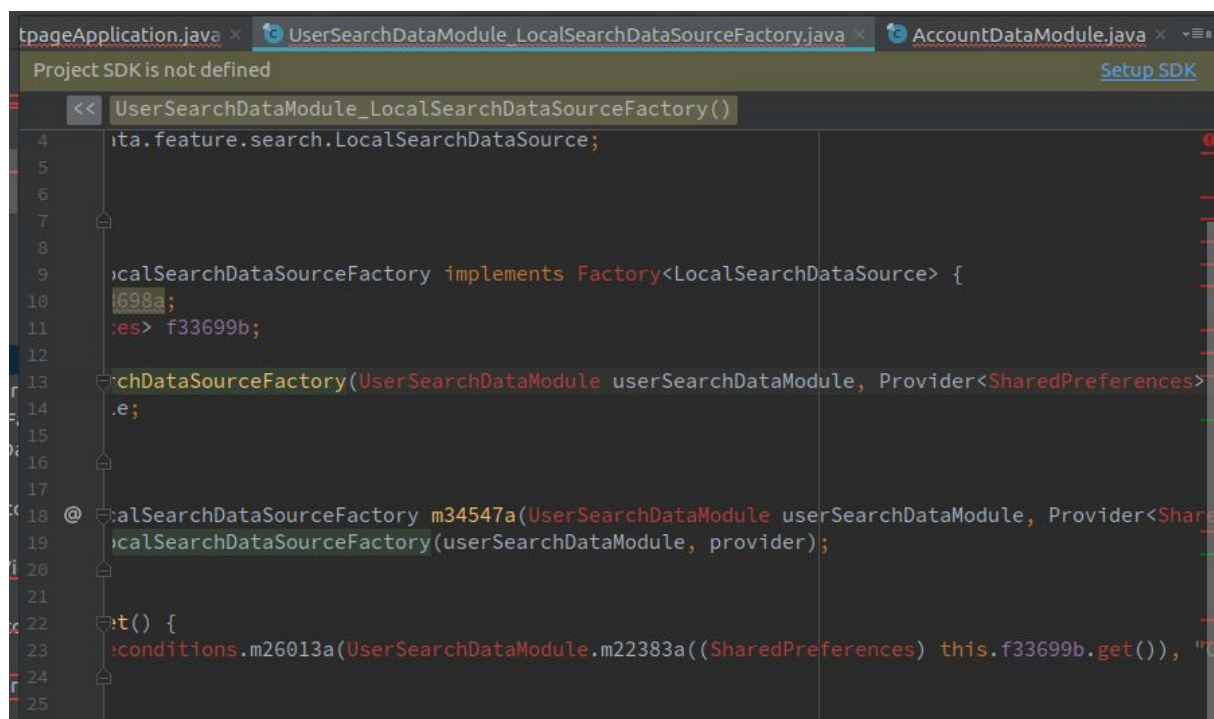
```
AccountDataModule
package com.redd.it.frontpage.di.module;

import ...

@Metadata(bv = {1, 0, 2}, d1 = {"\u00004\n\u0002\u0018\u0002\n\u0002\u0010\u0000\u0000\n\u0002\u0002\b"},
/* compiled from: AccountDataModule.kt */
public final class AccountDataModule {
    @Singleton
    public static RemoteAccountDataSource m22311a(Retrofit retrofit) {
        Intrinsic.m26847b(retrofit, "client");
        Object a = retrofit.a(RemoteAccountDataSource.class);
        Intrinsic.m26843a(a, "client.create(RemoteAccountDataSource::class.java)");
        return (RemoteAccountDataSource) a;
    }

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

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

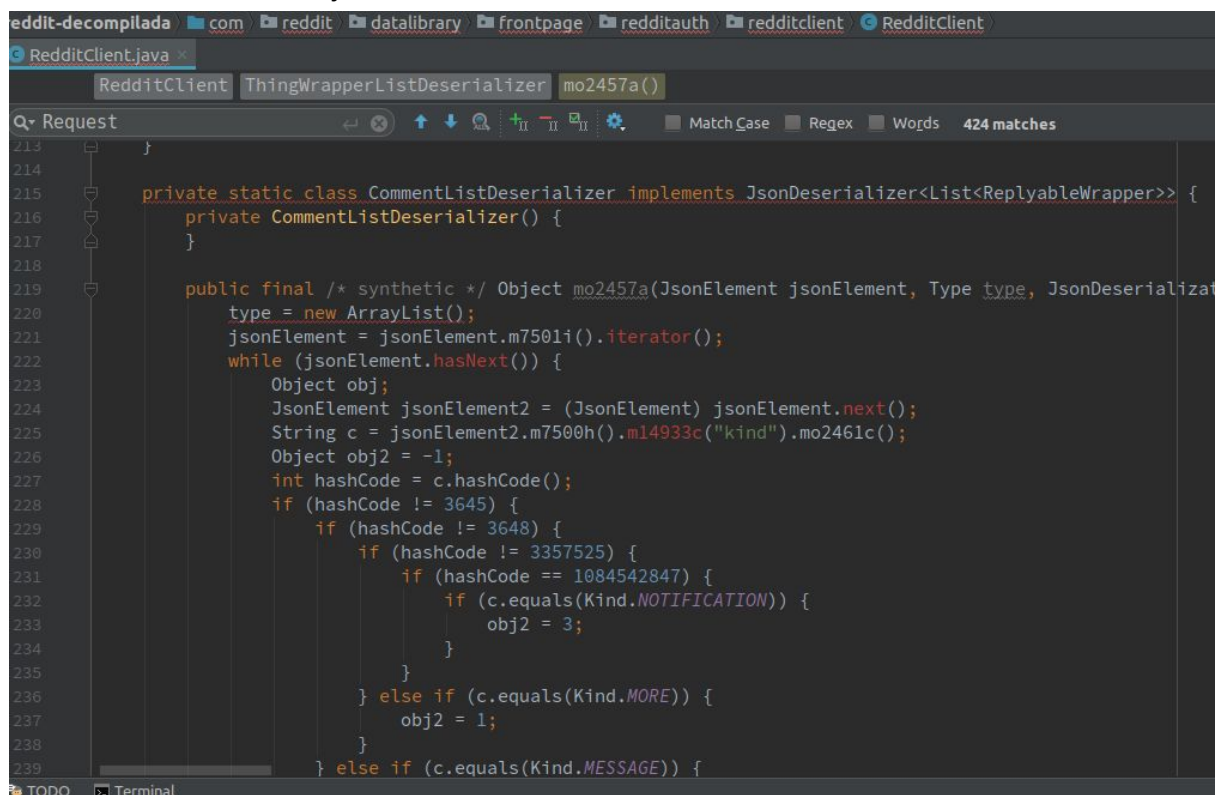
```
Project SDK is not defined Setup SDK

<< UserSearchDataModule_LocalSearchDataSourceFactory()

4 ita.feature.search.LocalSearchDataSource;
5
6
7
8
9 localSearchDataSourceFactory implements Factory<LocalSearchDataSource> {
10     f698a;
11     f33699b;
12
13     LocalSearchDataSourceFactory(UserSearchDataModule userSearchDataModule, Provider<SharedPreferences> provider);
14
15
16
17
18 @ LocalSearchDataSourceFactory m34547a(UserSearchDataModule userSearchDataModule, Provider<SharedPreferences> provider);
19     LocalSearchDataSourceFactory(userSearchDataModule, provider);
20
21
22     LocalSearchDataSourceFactory() {
23         LocalSearchDataSourceFactory(UserSearchDataModule.m22383a((SharedPreferences) this.f33699b.get()), "LocalSearchDataSourceFactory");
24     }
25 }
```

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()

Request 424 matches

213 }
214
215 private static class CommentListDeserializer implements JsonDeserializer<List<ReplyableWrapper>> {
216     private CommentListDeserializer() {
217     }
218
219     public final /* synthetic */ Object mo2457a(JsonElement jsonElement, Type type, JsonDeserializationContext context) throws JsonParseException {
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                         } else if (c.equals(Kind.MESSAGE)) {
238                             obj2 = 2;
239                         }
240                     }
241                 }
242             }
243             obj = new ReplyableWrapper(jsonElement2, obj2);
244             type.add(obj);
245         }
246         return type;
247     }
248 }
```

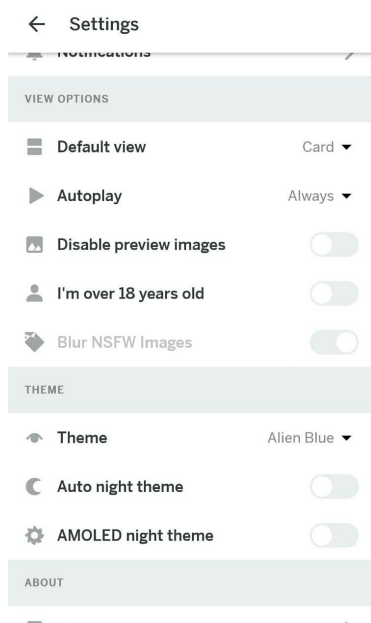
Estrategias de almacenamiento

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

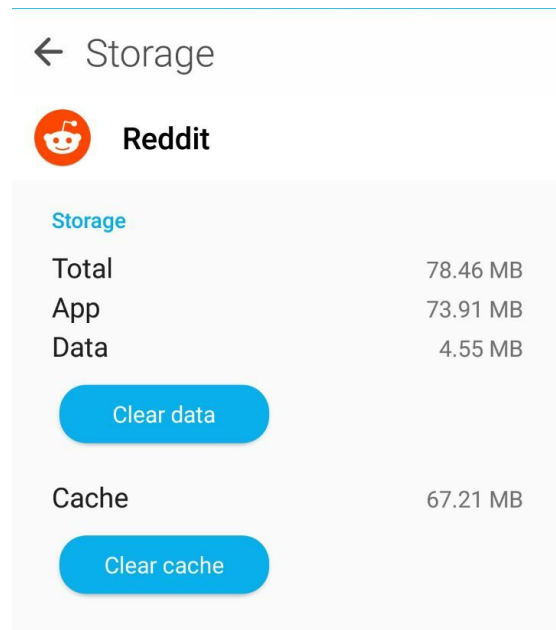
```
}
if (i <= 20500) {
    i = FrontpageApplication.f27402a.getSharedPreferences("com.reddit.frontpage.settings", 0);
    FrontpageSettings a2 = FrontpageSettings.a();
    a2.a(i.getBoolean("com.reddit.mobile.disable_preview_images", false));
    a2.a(i.getBoolean("com.reddit.pref.compact_listing_view", 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

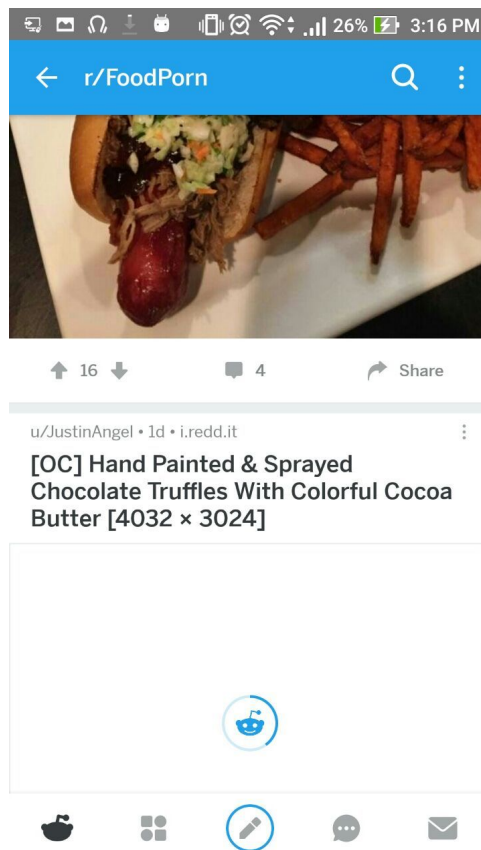
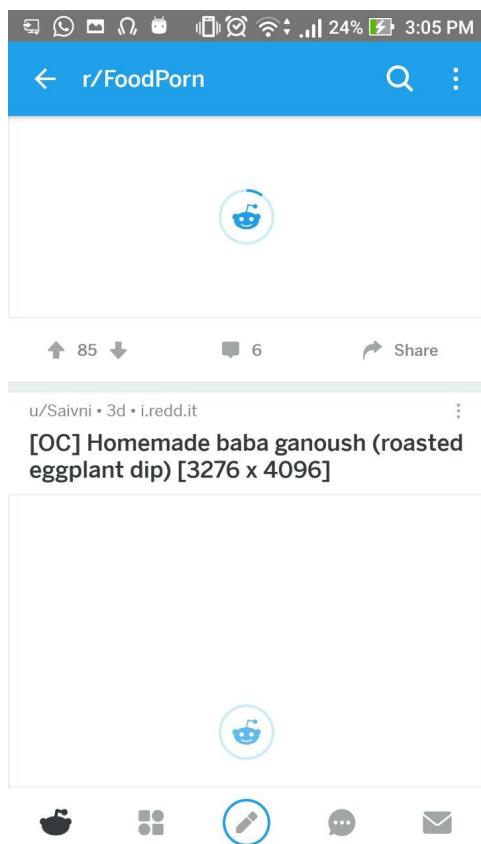
La aplicación presenta un gran consumo de almacenamiento y guarda mucha información en disco. Esto fue comprobado cuando al reiniciar el celular en modo sin conexión, todavía se podían visualizar los post guardados la última vez que esta se abrió.



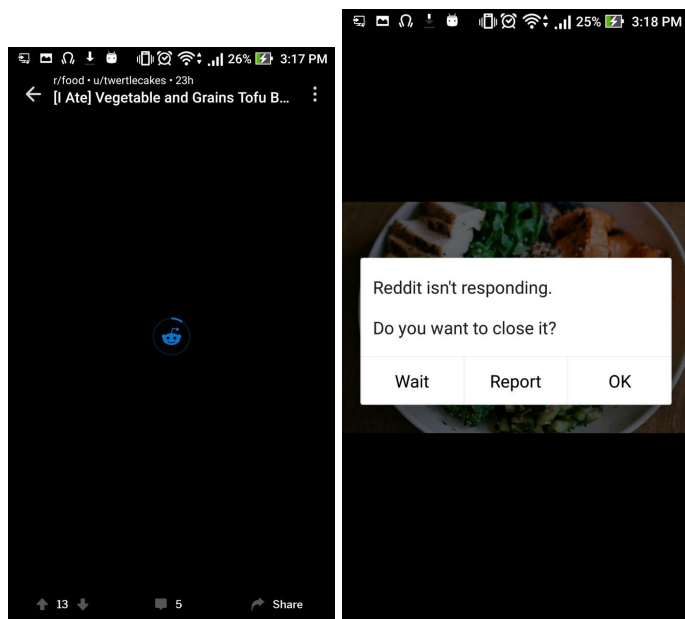
UI-Usabilidad

Manual

- Múltiples scroll a las publicaciones: solo carga/trae la publicación que está en pantalla



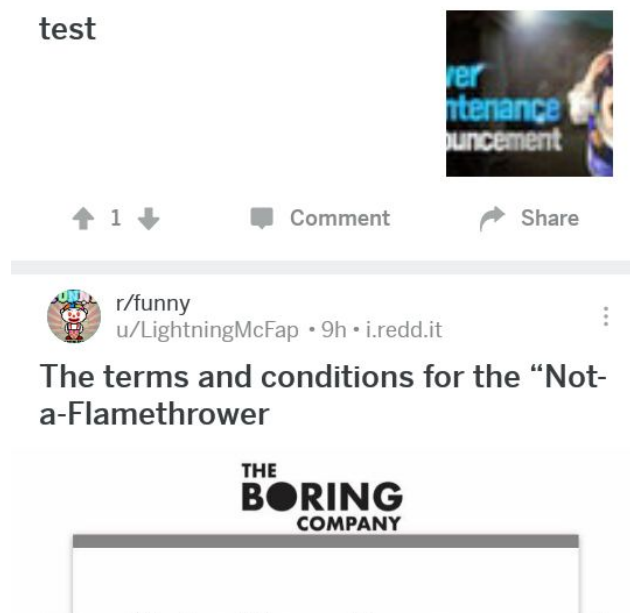
- Abrir la imagen de una publicación: tiempo de carga imagen



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.

