

Engenharia Reversa - Project 1 - Android Reversing

Date: 26/03/2023

Authors:

Camila Fonseca - 97880

Rodrigo Lima - 98475

Introduction

This report aims to present the findings of the reverse engineering of two mobile Android applications: the CTT app and the O Meu Pingodoce app. The project has the goal of analyzing the functionality, security, and overall design of these two popular applications.

The reverse engineering process involved both static and dynamic analysis techniques. Static analysis was used to examine the application's decompiled code, binaries, and resources, while dynamic analysis involved the use of various tools to observe the behavior of the applications in real-time.

Through this analysis, we aimed to identify any potential security vulnerabilities, performance issues, or design flaws in the applications. Additionally, we sought to gain a better understanding of the functionality of each application and how they interact with the users.

The CTT app and the Pingodoce app were chosen due to their widespread use and popularity in Portugal. By conducting this reverse engineering project, we hope to contribute to the improvement of these applications and to enhance the overall user experience for their users.

In the following sections, we will provide a detailed analysis of each application, including the tools and techniques used for analysis, the vulnerabilities and flaws identified, and recommendations for improvement.

CTT

Static Analysis

Android Manifest

Activities

The first step in our static analysis of the CTT app involved examining its manifest file. The manifest provides important information about the application, such as its package name, version number, and the permissions it requires to run.

This file is always present in the .apk (in some way or another), since it is needed by the android system for the application to work, which makes it a valuable entrypoint to gain understanding of the application. Particularly so due to Android's inner workings, as the app's interactions with the rest of the system (Access to hardware, event listeners...) must be detailed in the file.

During our analysis, we identified a launcher activity defined in the manifest file. The launcher activity in the CTT app is

`ctt.mobile.android.app.ctt.MainActivity`.

This activity is declared in the Android manifest file and has an intent filter with the `android.intent.action.MAIN` action and `android.intent.category.LAUNCHER` category, which makes it the entry point for the application when opened by its' Launcher (Its icon on the home screen is clicked on by the user).

The `android:exported` attribute is set to `true`, which means that it can be launched by other applications or components.

```
<activity
    android:name="ctt.mobile.android.app.ctt.MainActivity"
    android:label="@string/activity_name"
    android:theme="@style/Theme.OutSystems"
    android:launchMode="singleTop"
    android:screenOrientation="portrait"

    android:configChanges="smallestScreenSize|screenSize|uiMode|screenLayout|orientation|keyboardHidden|keyboard|locale"
    android:windowSoftInputMode="adjustResize"
    android:exported="true">

    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

```

        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="ctt.mobile.android.app.ctt" />
    </intent-filter>

</activity>

```

The `MainActivity` class extends `CordovaActivity`, which is a class provided by the Apache Cordova framework that enables developers to build mobile applications using HTML, CSS, and JavaScript. The `MainActivity` class overrides the `onCreate` method, which is called when the activity is created.

```

public class MainActivity extends CordovaActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Set by <content src="index.html" /> in config.xml
        loadUrl(this.launchUrl);
    }
}

```

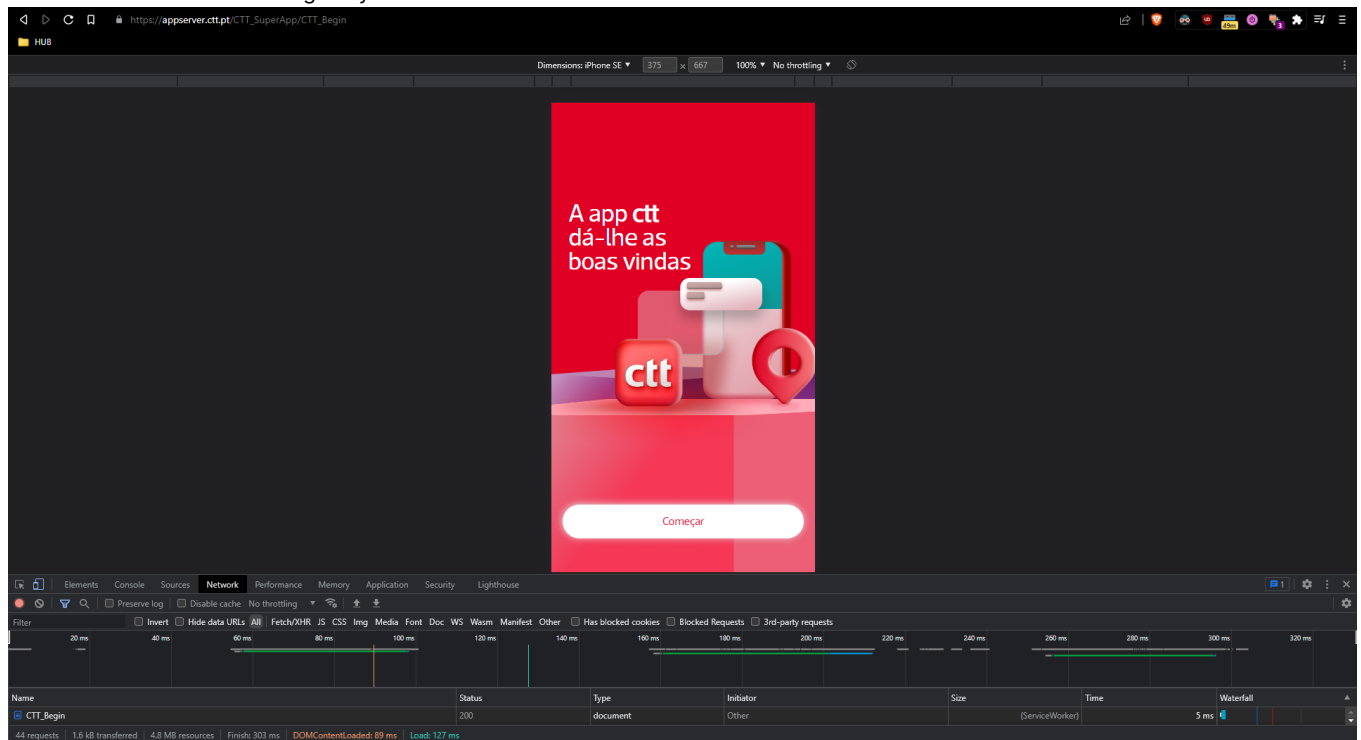
The `launchUrl` variable is defined and has its value set in `CordovaActivity`, and is imported from config preferences:

```

this.launchUrl = configXmlParser.getLaunchUrl();

```

The `loadUrl` method in the launcher activity starts a Cordova application based on JS/HTML, which is outside the scope of this project's analysis. However, it was possible to identify the application's URL as `https://appserver.ctt.pt/CTT_SuperApp/CTT_Begin`, which can be accessed through any modern browser.



Broadcast Receivers

Furthermore, we identified several Broadcast Receivers being used by the CTT app, which are capable of intercepting intents, and as such, be misused if exported, meaning it can intercept intents emitted by other applications.

One of them is the `CurrentAccessTokenExpirationBroadcastReceiver`.

```

<receiver android:name="com.facebook.CurrentAccessTokenExpirationBroadcastReceiver" android:exported="false">
    <intent-filter>
        <action android:name="com.facebook.sdk.ACTION_CURRENT_ACCESS_TOKEN_CHANGED"/>
    </intent-filter>
</receiver>

```

We found that this receiver was not exported, meaning it cannot be accessed by other applications or components.

Additionally, we found that the `CampaignTrackingReceiver` used by the CTT app was exported. While this receiver is a Facebook utility and not directly related to the CTT app's functionality, it can still be used as an entry point for analyzing the application.

The code defined in `com.facebook.CampaignTrackingReceiver` receives the `"com.android.vending.INSTALL_REFERRER"` intent.

```
public final class CampaignTrackingReceiver extends BroadcastReceiver {
    static final String INSTALL_REFERRER = "referrer";

    @Override // android.content.BroadcastReceiver
    public void onReceive(Context context, Intent intent) {
        String stringExtra = intent.getStringExtra("referrer");
        if (stringExtra == null || !stringExtra.startsWith("fb")) {
            return;
        }
        AppEventsLogger.setInstallReferrer(intent.getStringExtra("referrer"));
    }
}
```

This intent should have a `referrer` field, which must be a string starting with `fb`. Then, the `InstallReferrer` will be set to what's in the `referrer` value.

This `InstallReferrer` is later accessed by this method `getInstallReferrer`, defined in `com.facebook.appevents.AppEventsLoggerImpl`:

```
/* JADX INFO: Access modifiers changed from: package-private */
public static String getInstallReferrer() {
    return FacebookSdk.getApplicationContext()
        .getSharedPreferences("com.facebook.sdk.appEventPreferences",
            0).getString("install_referrer", null);
}
```

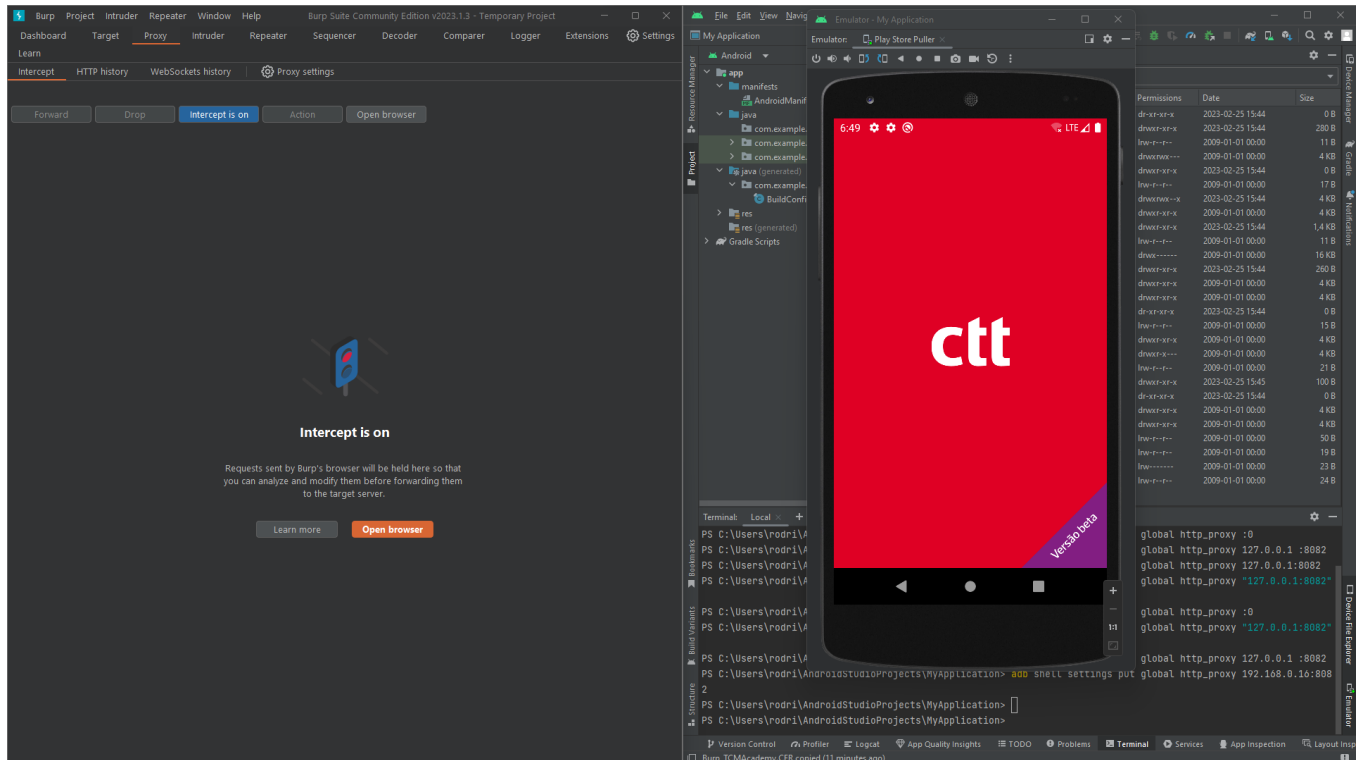
This is called only by the method `buildRequestForSession`, defined in `com.facebook.appevents.AppEventQueue`, which builds a Post request, later sent to the server in `sendEventsToServer` (in the same package).

This seems to be log/analytics for Facebook, and `installReferrer` is one of the parameters sent to the server as part of the post request.

Despite this value being seemingly easy to alter, by broadcasting an intent and following the restrictions mentioned above, the only effect this has is changing a bit of the logging information sent to the servers, and nothing else.

Dynamic Analysis

During the dynamic analysis of the CTT app, we encountered some challenges due to SSL pinning implemented by the app. This prevented us from intercepting traffic using the certificate we installed on the testing emulator.



To overcome this issue, we turned to Frida, an open-source dynamic instrumentation toolkit, which allows us to bypass SSL Pinning.

After installing and setting up Frida, we used an SSL unpinning script to intercept the app's traffic using BurpSuite.

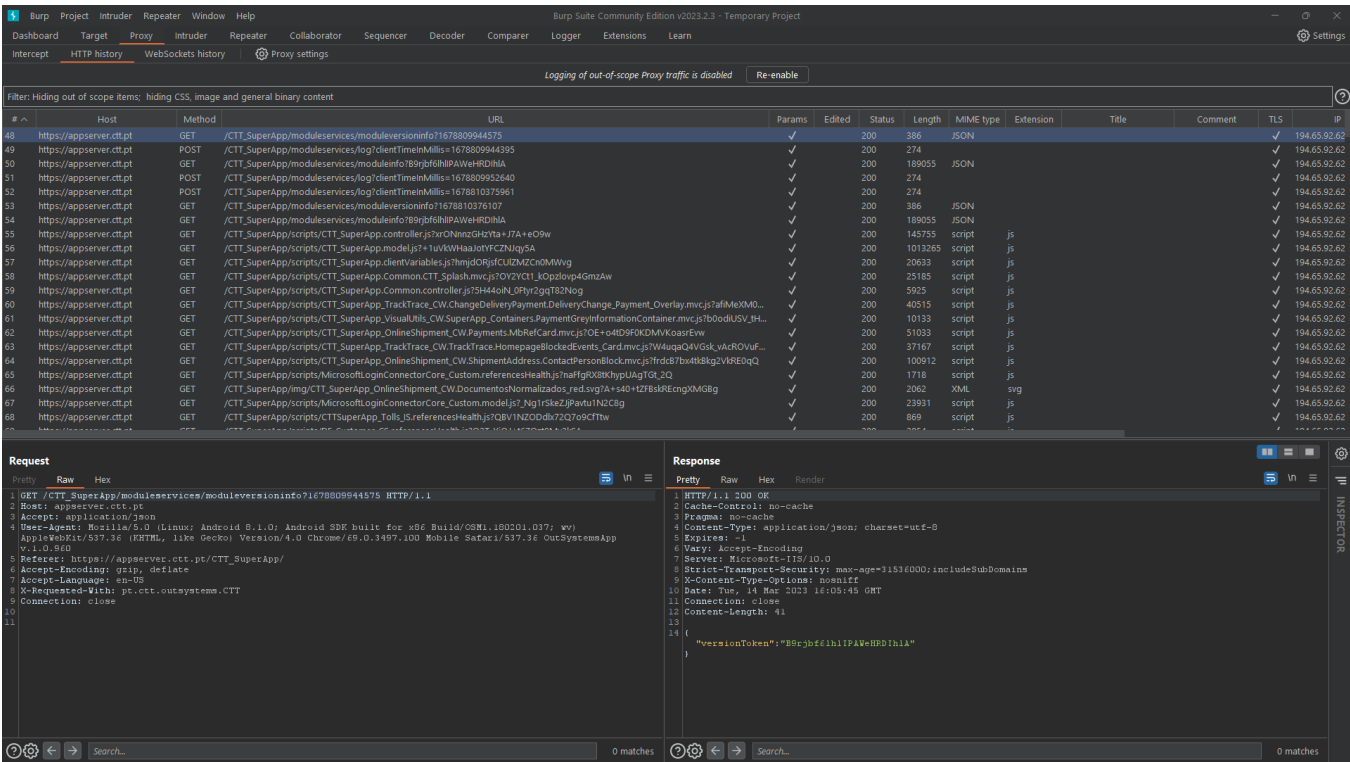
```

PS > adb push .\frida_multiple_unpinning.js /data/local/tmp
PS > adb shell pm list packages | findstr ctt
package:pt.ctt.outsystems.CTT
PS > frida -U -l frida_multiple_unpinning.js -f pt.ctt.outsystems.CTT

----
/ _ |   Frida 16.0.10 - A world-class dynamic instrumentation toolkit
| (| |
> _ |   Commands:
/_/ |_|   help      -> Displays the help system
. . . .   object?    -> Display information about 'object'
. . . .   exit/quit  -> Exit
. . . .
. . . .   More info at https://frida.re/docs/home/
. . . .
. . . .   Connected to Android Emulator 5554 (id=emulator-5554)
Spawned `pt.ctt.outsystems.CTT`. Resuming main thread!
[Android Emulator 5554:pt.ctt.outsystems.CTT ]->
=====
[#] Android Bypass for various Certificate Pinning methods [#]
=====
[-] OkHTTPv3 {2} pinner not found
[-] Trustkit {1} pinner not found
[-] Trustkit {2} pinner not found
[-] Trustkit {3} pinner not found
[-] Appcelerator PinningTrustManager pinner not found
[-] Fabric PinningTrustManager pinner not found
[-] OpenSSLSocketImpl Conscrypt {1} pinner not found
[-] OpenSSLSocketImpl Conscrypt {2} pinner not found
[-] OpenSSLEngineSocketImpl Conscrypt pinner not found
[-] OpenSSLSocketImpl Apache Harmony pinner not found
[-] PhoneGap sslCertificateChecker pinner not found
[-] IBM MobileFirst pinTrustedCertificatePublicKey {1} pinner not found
[-] IBM MobileFirst pinTrustedCertificatePublicKey {2} pinner not found
[-] IBM WorkLight HostNameVerifierWithCertificatePinning {1} pinner not found
[-] IBM WorkLight HostNameVerifierWithCertificatePinning {2} pinner not found
[-] IBM WorkLight HostNameVerifierWithCertificatePinning {3} pinner not found
[-] IBM WorkLight HostNameVerifierWithCertificatePinning {4} pinner not found
[-] Conscrypt CertPinManager (Legacy) pinner not found
[-] CWAC-Netsecurity CertPinManager pinner not found
[-] Worklight Androidgap WLCertificatePinningPlugin pinner not found
[-] Netty FingerprintTrustManagerFactory pinner not found
[-] Squareup CertificatePinner {1} pinner not found
[-] Squareup CertificatePinner {2} pinner not found
[-] Squareup OkHostnameVerifier check not found
[-] Squareup OkHostnameVerifier check not found
[-] Android WebViewClient {2} check not found
[-] Apache Cordova WebViewClient check not found
[-] Boye AbstractVerifier check not found
[-] Apache AbstractVerifier check not found
[-] Chromium Cronet pinner not found
[-] Flutter HttpCertificatePinning pinner not found
[-] Flutter SslPinningPlugin pinner not found
[+] Bypassing Trustmanager (Android < 7) pinner
[+] Bypassing Trustmanager (Android < 7) pinner
[+] Bypassing Trustmanager (Android < 7) pinner
[+] Bypassing Android WebViewClient check {4}
[+] Bypassing Trustmanager (Android < 7) pinner
[+] Bypassing TrustManagerImpl (Android > 7) checkTrustedRecursive check: appserver.ctt.pt
[+] Bypassing OkHTTPv3 {4}: appserver.ctt.pt
[+] Bypassing OkHTTPv3 {4}: appserver.ctt.pt
[+] Bypassing OkHTTPv3 {4}: appserver.ctt.pt
[+] Bypassing OkHTTPv3 {4}: appserver.ctt.pt
[+] Bypassing OkHTTPv3 {4}: appserver.ctt.pt
[+] Bypassing OkHTTPv3 {4}: appserver.ctt.pt

```

This enabled us to see and analyze network traffic, including requests and responses sent and received by the app. We could examine the HTTP headers, request and response bodies, and identify the APIs called by the app, among others.



Through this process, we discovered all the HTML and JavaScript requested by the app. This was made possible by the Cordova technology. We intercepted the app's network traffic and analyzed the requests and responses exchanged between the app and the server. This allowed us to identify the different endpoints used by the app and the data being sent and received by the app.

PingoDoce

This is the second application analysed in this report, and can be found in the Play Store with the name **0 Meu PingoDoce**.

Static Analysis

As with the previous application, the APK was retrieved from Google's Play Store in an Android Emulator and decompiled using Jadx 1.4.6.

Manifest Analysis

Exported Activities

RecoverPinActivity

Manifest Snippet:

```
<activity android:name="pt.pingodoce.app.presentation.login.pin.change.RecoverPinActivity" android:exported="true"
android:launchMode="singleTask"
android:screenOrientation="portrait">
    <intent-filter android:label="@string/app_name_label">
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE"/>
        <data android:scheme="@string/deeplink_scheme_pin" android:host="@string/deeplink_host"
android:pathPrefix="@string/deeplink_prefix_pin"/>
    </intent-filter>
</activity>
```

In this activity, the only code related to intents is the following:

```
@Override // id.Functions
public final AbstractSavedStateViewModelFactory acceptRGPDTermsNConditionsDialog() {
    ViewModelFactoryByInjection ifNotNull = RecoverPinActivity
        .this.getIfNotNull();
    RecoverPinActivity recoverPinActivity = RecoverPinActivity.this;
    Intent intent = recoverPinActivity.getIntent();
    return ifNotNull.m18579b(
        recoverPinActivity,
```

```

        intent != null ? intent.getExtras() : null);
    }

```

Unfortunately the function `m18579b` (`b()` below) couldn't be reversed.

```

public final androidx.lifecycle.a b(androidx.savedstate.c activity, Bundle intent) {
    Intrinsic.isThisObjectNull(activity, "owner");
    return new a(activity, intent, this);
}

public static final class a extends androidx.lifecycle.a {

    final /* synthetic */ d f14321d;

    a(androidx.savedstate.c cVar, Bundle bundle, d dVar) {
        super(cVar, bundle);
        this.f14321d = dVar;
    }

    @Override // androidx.lifecycle.a
    protected <T extends o0> T d(String str, Class<T> cls, l0 l0Var) {
        Intrinsic.isThisObjectNull(str, "key");
        Intrinsic.isThisObjectNull(cls, "modelClass");
        Intrinsic.isThisObjectNull(l0Var, "handle");
        b bVar = (b) this.f14321d.f14320a.get(cls);
        T t10 = bVar != null ? (T) bVar.a(l0Var) : null;
        if (t10 != null) {
            return t10;
        }
        throw new IllegalStateException("Unknown ViewModel class");
    }
}

```

As such, no further analysis was possible.

Launcher - Splash Activity

During the static analysis of the Pingodoce app, the launcher activity `SplashActivity` was identified as the one that is launched when a user opens the application via its homescreen icon. This was confirmed by analyzing the manifest file of the app.

```

<activity android:theme="@style/App.Splash"
    android:name="pt.pingodoce.app.presentation.landing.splash.SplashActivity"
    android:exported="true"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>

```

As this activity has set an intent-filter for `LAUNCHER`, this is the activity that is ran when a user opens the application via its homescreen icon.



Note

In the `onCreate` method of `SplashActivity`, there is a check for the presence of an 'extra' field called `E-GOI-PUSH` in the intent. If this field is present, the app tries to launch a new activity called `NotificationLandingActivity` with the intent as an argument. Decompiled and cleaned up code:

```

@Override
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    if (getIntent().hasExtra("E-GOI-PUSH")) {
        Intent intent = getIntent();
    }
}

```

```

        Intrinsic.checkNotNull(intent, "intent");
        NotificationLandingActivity.
            startActivityWithIntentIfNotNull(this, intent);
        return;
    }

```

This code starts a new activity. Due to the structure of the intent's constructor, this seems to be the types used:

```
Intent(Context packageContext, Class<?> cls)
```

Create an intent for a specific component.

(Source: Google's [Developer Docs](#))

Here, the Context has this Class signature: `pt.pingodoce.app.services.NotificationLandingActivity.class`

After launching the application, Pingodoce's app checks if it is running with root permissions or if it is running inside a virtual machine.

The code is checking for the presence of root on the device. It uses the [RootBeer](#) library to perform the check. The check is called from the `ai.m` class and it resides on the class `nc.b`:

```

package nc;

<SNIP>

/* compiled from: RootBeer.java */
/* loaded from: classes2.dex */
public class b {
    <SNIP>

```

The `RootBeer` library has a set of methods that perform various checks to determine whether the device is rooted. The library uses a combination of checks to determine whether a device is rooted or not. Some of the checks include:

- Checking for the presence of specific files and directories that are present on rooted devices.
- Checking if any of the installed apps on the device have been granted root access.
- Checking if the device has any of the known root management apps installed.
- Checking if the `su` binary is present on the device.
- Checking if the device is in debug mode.

If any of these checks succeed, then the device is considered rooted. The `RootBeer` library returns a boolean value indicating whether the device is rooted or not.

The `ai.m` class invokes the `a()` method of the `RootBeer` library to perform the root check.

```

@Override // ai.n
public void a() {
    if (this.f323c.o()) {
        new b.a(this.f321a,
2131951632).p(R.string.alert_device_has_root_title).g(R.string.alert_device_has_root_message).i(g() ?
R.string.alert_device_has_root_exit : R.string.alert_device_has_root_exit_alt, new DialogInterface.OnClickListener()
{ // from class: ai.l
    {
        m.this = this;
    }

    @Override // android.content.DialogInterface.OnClickListener
    public final void onClick(DialogInterface dialogInterface, int i10) {
        m.e(m.this, dialogInterface, i10);
    }
}).d(false).a().show();
    }
    <SNIP>
}

```

If the device is rooted, then a dialog is displayed to the user with the message `"Dispositivo não seguro"` indicating that the device is rooted. The user is also given an option to proceed.

This check can be bypassed by the use of the Frida tool with the help of [RootCheckBypass.js](#)

```
PS > frida -U -l .\RootCheckBypass.js -f pt.pingodoce
```

```

/ _ |   Frida 16.0.10 - A world-class dynamic instrumentation toolkit
| (| |
> _ |   Commands:
/_/ |_|   help      -> Displays the help system
. . . .   object?    -> Display information about 'object'
. . . .   exit/quit -> Exit
. . . .
. . . .   More info at https://frida.re/docs/home/
. . . .
. . . .   Connected to Android Emulator 5554 (id=emulator-5554)
Spawned `pt.pingodoce`. Resuming main thread!
<SNIP>
[Android Emulator 5554::pt.pingodoce ]->

```

This indicates a potential security issue as a malicious user with root access to their device could also bypass this check and potentially exploit the app's vulnerabilities. It's recommend to impre the root detection mechanism to prevent such bypasses.

The app also has a check in place to see if it's running inside a virtual machine (VM). This check is called on the `ai.m` class, specifically in the `a()` method. If the check determines that the app is running inside a VM, then a dialog is displayed to the user with the message `"Emulador detectado"` and an option to proceed.

```

@Override // ai.n
public void a() {
    if (this.f323c.o()) {
        <SNIP>
    } else if (h()) {
        new b.a(this.f321a,
2131951632).p(R.string.alert_device_is_emulator_title).g(R.string.alert_device_is_emulator_message).i(g() ?
R.string.alert_device_is_emulator_exit : R.string.alert_device_is_emulator_exit_alt, new
DialogInterface.OnClickListener() { // from class: ai.k
    {
        m.this = this;
    }

    @Override // android.content.DialogInterface.OnClickListener
    public final void onClick(DialogInterface dialogInterface, int i10) {
        m.f(m.this, dialogInterface, i10);
    }
}).d(false).a().show();
    }
}
}

```

The relevant code for this check can be seen in the provided (decompiled) Java snippet:

```

private final boolean h() {
    r10 = this;
    java.lang.String r0 = android.os.Build.FINGERPRINT
    java.lang.String r1 = "FINGERPRINT"
    jd.l.g(r0, r1)
    java.lang.String r2 = "google/sdk_gphone_"
    r3 = 0
    r4 = 2
    r5 = 0
    boolean r2 = sd.k.C(r0, r2, r3, r4, r5)
    java.lang.String r6 = "PRODUCT"
    java.lang.String r7 = "MODEL"
    if (r2 == 0) goto L4d
    jd.l.g(r0, r1)
    java.lang.String r2 = ":user/release-keys"
    boolean r2 = sd.k.p(r0, r2, r3, r4, r5)
    if (r2 == 0) goto L4d
    java.lang.String r2 = android.os.Build.MANUFACTURER
    java.lang.String r8 = "Google"
    boolean r2 = jd.l.c(r2, r8)
    if (r2 == 0) goto L4d
    java.lang.String r2 = android.os.Build.PRODUCT
    jd.l.g(r2, r6)
    java.lang.String r8 = "sdk_gphone_"
    boolean r2 = sd.k.C(r2, r8, r3, r4, r5)
    if (r2 == 0) goto L4d
    java.lang.String r2 = android.os.Build.BRAND
    java.lang.String r9 = "google"
    boolean r2 = jd.l.c(r2, r9)
}

```



```

        if (r2 == 0) goto L4d
        java.lang.String r2 = android.os.Build.MODEL
        jd.l.g(r2, r7)
        boolean r2 = sd.k.C(r2, r8, r3, r4, r5)
        if (r2 != 0) goto Lcd

L4d:
        jd.l.g(r0, r1)
        java.lang.String r2 = "generic"
        boolean r8 = sd.k.C(r0, r2, r3, r4, r5)
        if (r8 != 0) goto Lcd
        jd.l.g(r0, r1)
        java.lang.String r1 = "unknown"
        boolean r0 = sd.k.C(r0, r1, r3, r4, r5)
        if (r0 != 0) goto Lcd
        java.lang.String r0 = android.os.Build.MODEL
        jd.l.g(r0, r7)
        java.lang.String r1 = "google_sdk"
        boolean r8 = sd.k.H(r0, r1, r3, r4, r5)
        if (r8 != 0) goto Lcd
        jd.l.g(r0, r7)
        java.lang.String r8 = "Emulator"
        boolean r8 = sd.k.H(r0, r8, r3, r4, r5)
        if (r8 != 0) goto Lcd
        jd.l.g(r0, r7)
        java.lang.String r7 = "Android SDK built for x86"
        boolean r0 = sd.k.H(r0, r7, r3, r4, r5)
        if (r0 != 0) goto Lcd
        java.lang.String r0 = android.os.Build.MANUFACTURER
        java.lang.String r7 = "MANUFACTURER"
        jd.l.g(r0, r7)
        java.lang.String r8 = "VMware"
        boolean r8 = sd.k.H(r0, r8, r3, r4, r5)
        if (r8 != 0) goto Lcd
        jd.l.g(r0, r7)
        java.lang.String r7 = "Genymotion"
        boolean r0 = sd.k.H(r0, r7, r3, r4, r5)
        if (r0 != 0) goto Lcd
        java.lang.String r0 = android.os.Build.BRAND
        java.lang.String r7 = "BRAND"
        jd.l.g(r0, r7)
        boolean r0 = sd.k.C(r0, r2, r3, r4, r5)
        if (r0 == 0) goto Lba
        java.lang.String r0 = android.os.Build.DEVICE
        java.lang.String r7 = "DEVICE"
        jd.l.g(r0, r7)
        boolean r0 = sd.k.C(r0, r2, r3, r4, r5)
        if (r0 != 0) goto Lcd

Lba:
        java.lang.String r0 = android.os.Build.PRODUCT
        boolean r1 = jd.l.c(r0, r1)
        if (r1 != 0) goto Lcd
        jd.l.g(r0, r6)
        java.lang.String r1 = "vbox86p"
        boolean r0 = sd.k.H(r0, r1, r3, r4, r5)
        if (r0 == 0) goto Lce

Lcd:
        r3 = 1

Lce:
        return r3

throw new UnsupportedOperationException("Method not decompiled: ai.m.h():boolean");
}

```

The check for running inside a VM can be bypassed using a Frida script. The script hooks the `h()` method of the `ai.m` class as shown below:

```

let m = Java.use("ai.m"); m["h"].implementation = function () {
    console.log(`m.h is called`);
    let result = this["h"]();
    console.log(`Tampering result: ${result}`);
    return false;
};

```

Once the script is executed, it replaces the original implementation of the `h()` method with a custom implementation that always returns `false`. As a result, the app is fooled into believing that it is not running inside a VM.

```
PS C:\Users\rodri\Desktop\ER\Frida> frida -U -l '.\Root&VMCheckBypass.js' -f pt.pingodoce

----
/ _ |   Frida 16.0.10 - A world-class dynamic instrumentation toolkit
| (| |
> _ |   Commands:
/_/ |_|   help       -> Displays the help system
. . . .   object?    -> Display information about 'object'
. . . .   exit/quit  -> Exit
. . . .
. . . .   More info at https://frida.re/docs/home/
. . . .
. . . .   Connected to Android Emulator 5554 (id=emulator-5554)
Spawned `pt.pingodoce`. Resuming main thread!
<SNIP>
m.h is called
Tampering result: true
```

When the user clicks on the "OK FUI AVISADO" button and the root or VM check fails, a promotional screen showcasing the app's functionalities is displayed.



By searching for a specific text displayed on the screen in Jadx, it is possible to locate the corresponding string resource that contains the text. In this case, the string resource for the text `Os meus folhetos` can be found with the name `"lbl_walkthrough_step1"` in `res/values/strings.xml`.

This string resource is then used in `res/layout/fragment_walkthrough_step1.xml`, which is inflated in the `AppWalkThroughStep1Fragment` class extending `WalkThroughFragment`. By searching for the name of this class, we can find the `AppWalkThroughActivity`.

This activity can then be traced back to `SplashActivity` using the following code (slightly obfuscated):

```
if (Intrinsics.equals(c13182l.m1459d(), Boolean.TRUE)) {
    AppWalkThroughActivity.m22601a(SplashActivity.this);
}
```

After some code analysis and renaming, the following code is obtained:

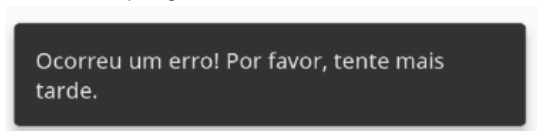
```
if (Intrinsics.equals(c13801l.getPossibleFirstLaunchBool(), Boolean.TRUE)) {  
    AppWalkThroughActivity.launchAppWalkThroughActivityIfNotNull(SplashActivity.this);  
}
```

After this walkthrough is either skipped or completed the user is greeted with a Welcome Page:



The page referred to as `IntroActivity` is encountered at this stage.

When attempting to click the 'ou entrar sem conta' button on the emulator, an error is generated:



This message can be traced back to the string resource `err_api_generic` which is then used to throw an API Error:

```
package af;  
  
import pl81jd.Intrinsics;  
  
/* compiled from: ApiError.kt */  
/* renamed from: af.a */  
/* loaded from: classes2.dex */  
public final class APIError {  
  
    /* renamed from: a */  
    private final String errorMessage;  
  
    public APIError(String str) {  
        Intrinsics.isThisObjectNull(str, "message");  
        this.errorMessage = str;  
    }  
  
    /* renamed from: a */  
    public final String getMessage() {  
        return this.errorMessage;  
    }  
  
    public boolean equals(Object obj) {  
        if (this == obj) {  
            return true;  
        }  
        return (obj instanceof APIError) && Intrinsics.equals(this.errorMessage, ((APIError) obj).errorMessage);  
    }  
}
```

```

}

public int hashCode() {
    return this.errorMessage.hashCode();
}

public String toString() {
    String str = this.errorMessage;
    return "ApiError(message=" + str + ")";
}
}

```

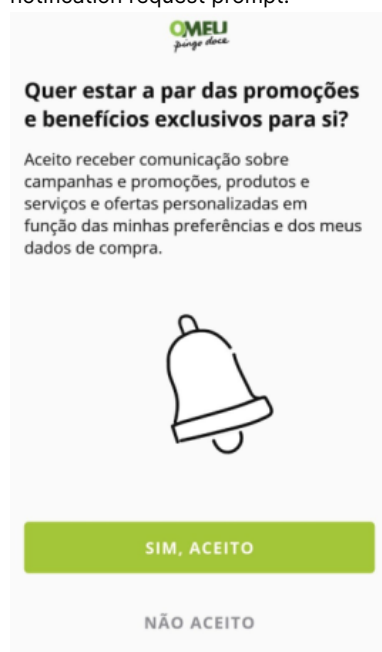
and this object is used in the method below:

```

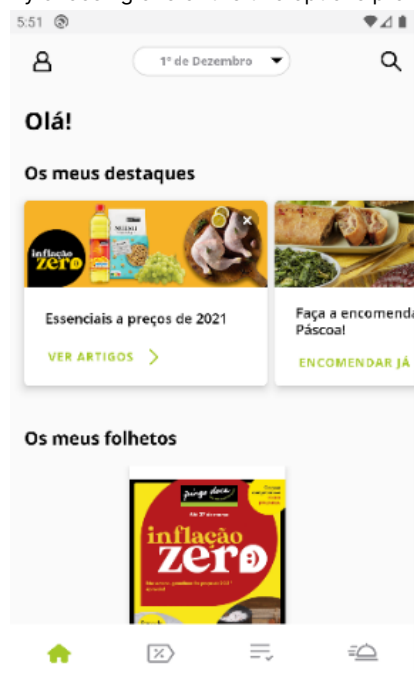
public final ApiError throwApiError(int i) {
    String string = this.f13109a.m6477a().getString(i);
    Intrinsics.checkNotNull(string, "ctx.getString(id)");
    return new ApiError(string);
}

```

However, no indication as to what went wrong is given with this approach. If the error isn't encountered, we're greeted with a notification request prompt.



By choosing one of the two options provided we were then directed to the Main Page.



The main activity associated with this page is `HomeActivity`, and during the first launch, the user is presented with a tutorial through coach marks. If one searches for the text present in the interface (which corresponds to the string resource `btn_store`), they can identify that it is part of `pt.pingodoce.app.presentation.home.HomeCoachMarks$showCoachMarksForLoyalty$2`. However, we will not delve into this any further here.

🔗 Error

During static analysis multiple parts of the code followed this pattern:

```
@kotlin.coroutines.jvm.internal.f(c =
"pt.pingodoce.app.presentation.home.HomeCoachMarks$showCoachMarksForLoyalty$2", f = "HomeCoachMarks.kt", l = {89,
91}, m = "invokeSuspend")
public static final class j extends kotlin.coroutines.jvm.internal.l implements id.p<i0, bd.d<? super yc.u>, Object>
{

    int f23216x;
    final /* synthetic */ e0 f23217y;

    j(e0 e0Var, bd.d<? super j> dVar) {
        super(2, dVar);
        this.f23217y = e0Var;
    }

    @Override // kotlin.coroutines.jvm.internal.a
    public final bd.d<yc.u> create(Object obj, bd.d<?> dVar) {
        return new j(this.f23217y, dVar);
    }

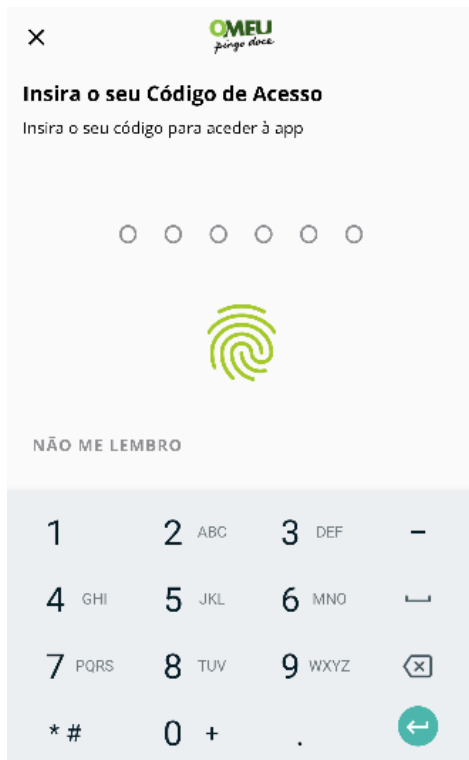
    @Override // id.p
    /* renamed from: d */
    public final Object F(i0 i0Var, bd.d<? super yc.u> dVar) {
        return ((j) create(i0Var, dVar)).invokeSuspend(yc.u.f27252a);
    }

    @Override // kotlin.coroutines.jvm.internal.a
    public final Object invokeSuspend(Object obj) {
        Object d10;
        d10 = cd.d.d();
        int i10 = this.f23216x;
        if (i10 == 0 || i10 == 1) {
            yc.n.b(obj);
            while (this.f23217y.C.getWidth() == 0) {
                this.f23216x = 1;
                if (r0.a(100L, this) == d10) {
                    return d10;
                }
            }
            this.f23216x = 2;
            if (r0.a(300L, this) == d10) {
                return d10;
            }
        } else if (i10 != 2) {
            throw new IllegalStateException("call to 'resume' before 'invoke' with coroutine");
        } else {
            yc.n.b(obj);
        }
        return yc.u.f27252a;
    }
}
```

🔗 Error

Due to the usage of these coroutines, analysis became much harder and as such these methods/classes were considered dead-ends. These will appear later on in this report and can be identified by the presence of '\$' in the method/class paths.

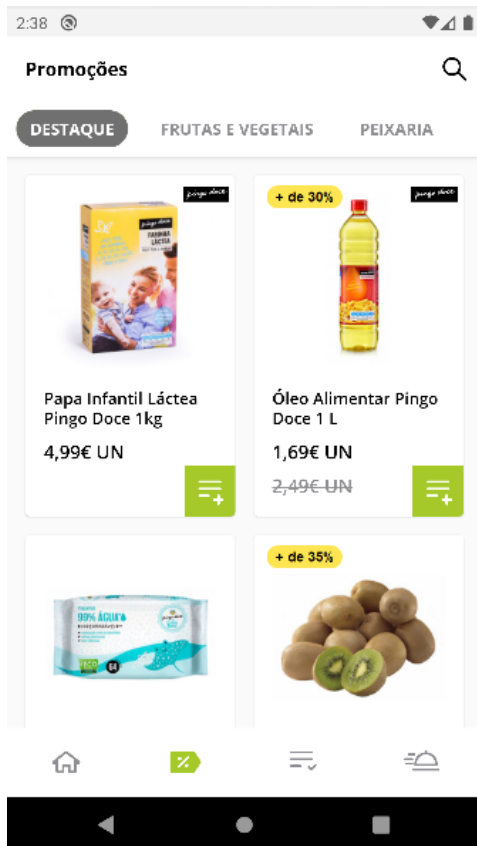
Unlock Screen



The method that handles the validation of the pin can be found in

`pt.pingodoce.app.presentation.login.phone.update.ValidatePinForUpdatePhoneViewModel$onPinCompleted$1$1`, which was not further analyzed.

Promotions Screen



The activity for this screen is `PromotionActivity`, and its package is `pt.pingodoce.app.presentation.promotion`.

Search

This is the code present in `PromotionActivity` that seems to handle search (then calling on `SearchActivity` methods):

```
public static final boolean A1(PromotionActivity promotionActivity,
                               MenuItem menuItem) {
```

```

Intrinsics.isThisObjectNull(promotionActivity, "this$0");
Integer valueOf = menuItem != null ?
                                Integer.valueOf(menuItem.getItemId()) :
                                null;

if (valueOf != null && valueOf.intValue() == R.id.action_search) {
    SearchActivityHelper.setExtrasOnIntentWrapper(
        promotionActivity.f20096j0, promotionActivity,
        Boolean.TRUE, null, 4, null);
    return true;
}
return false;
}

```

This code gets the value of a *certain* menu item ID, and then if it corresponds to `R.id.action_search` it will call methods from `SearchActivityHelper` (Further detailed ahead) in order to create an intent.

`R.id.action_search` is referenced in `Toolbar`, `ShoppingListActivity`, `HomeActivity` and `FlyerActivity`, so it likely is related to the search icon present in the toolbar in multiple screens.



Search Page/Function

SearchActivityHelper

This activity contains methods that are used to create/structure intents seemingly for the various search functions in the app.

From the [Promotions](#) page (`PromotionActivity`) the following decompiled methods are called:

```

public static /* synthetic */ void callSearchAfterMatch(androidx.activity.result.b bVar, Activity activity, Boolean
bool, Integer num, int i10, Object obj) {
    if ((i10 & 2) != 0) {
        bool = null;
    }
    if ((i10 & 4) != 0) {
        num = null;
    }
    a(bVar, activity, bool, num);
}

public static final void a(androidx.activity.result.b<Intent> bVar, Activity activity, Boolean bool, Integer num) {
    Intrinsics.isThisObjectNull(bVar, "<this>");
    Intrinsics.isThisObjectNull(activity, "context");
    Intent intent = new Intent(activity, pt.pingodoce.app.presentation.common.search.SearchActivity.class);
    if (bool != null) {
        bool.booleanValue();
        intent.putExtra("EXTRA_FOR_HAS_PROMO", bool.booleanValue());
    }
    if (num != null) {
        num.intValue();
        intent.putExtra("EXTRA_FOR_FLYER_ID", num.intValue());
    }
    bVar.a(intent);
}

```

and these can be cleaned up into:

```

public static final void setExtrasOnIntent(androidx.activity.result.b<Intent> intentWrapper, Activity activity,
Boolean bool, Integer num) {
    Intrinsics.isThisObjectNull(intentWrapper, "<this>");
    Intrinsics.isThisObjectNull(activity, "context");
    Intent intent = new Intent(activity, SearchActivity.class);
    if (bool != null) {
        bool.booleanValue();
        intent.putExtra("EXTRA_FOR_HAS_PROMO", bool.booleanValue());
    }
    if (num != null) {
        num.intValue();
        intent.putExtra("EXTRA_FOR_FLYER_ID", num.intValue());
    }
    intentWrapper.doSomethingWithIntent(intent);
}

```

```

public static void setExtrasOnIntentWrapper(androidx.activity.result.b bVar, Activity activity, Boolean bool,
Integer num, int sel, Object obj) {
    if (sel==2) {
        bool = null;
    }
    if (sel==4) {
        num = null;
    }
    setExtrasOnIntent(bVar, activity, bool, num);
}

```

So, this method will handle creation of an Intent, and depending on the value that is passed to `setExtrasOnIntentWrapper`, different Extra fields will be added to the Intent.

`EXTRA_FOR_HAS_PROMO` and `EXTRA_FOR_FLYER_ID` are later accessed in `SearchActivity`, and seen in the following code:

```

private final void U1() {
    ((pt.pingodoce.app.presentation.common.search.a) IO()).o1(getIntent().getBooleanExtra("EXTRA_FOR_HAS_PROMO",
false));
    int intExtra = getIntent().getIntExtra("EXTRA_FOR_FLYER_ID", -1);
    if (intExtra != -1) {
        ((pt.pingodoce.app.presentation.common.search.a) IO()).n1(intExtra);
    }
    String stringExtra = getIntent().getStringExtra("EXTRA_FOR_QUERY");
    if (stringExtra != null) {
        K1().d0(stringExtra, true);
    }
}

// Cleaned up

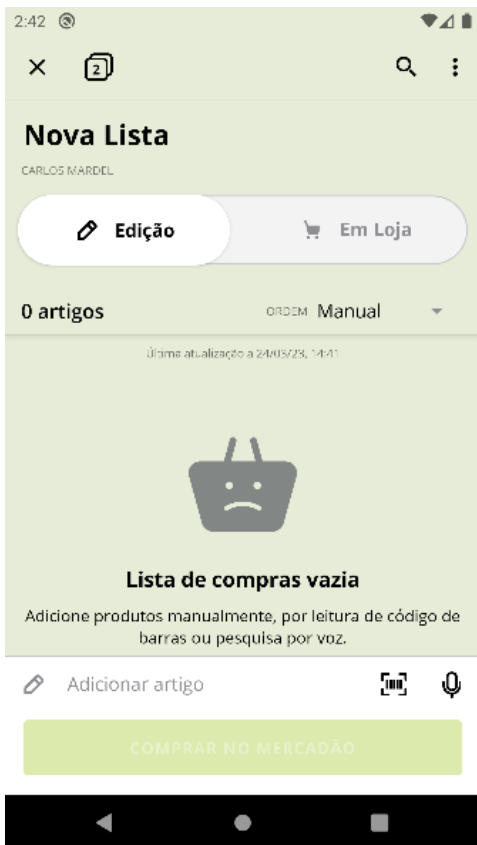
private final void handlePromoIntent() {
    aSearchViewModel().setPromo(
        getIntent().getBooleanExtra("EXTRA_FOR_HAS_PROMO", false));
    int intExtra = getIntent().getIntExtra("EXTRA_FOR_FLYER_ID", -1);
    if (intExtra != -1) {
        aSearchViewModel().setFlyerId(intExtra);
    }
    String stringExtra = getIntent().getStringExtra("EXTRA_FOR_QUERY");
    if (stringExtra != null) {
        aSearchView().setTextForAutoComplete(stringExtra, true);
    }
}

```

These Extra Intent fields seem to be used to determine which data to fill the search ViewModel and related components.

A similar method in this class does roughly the same for `EXTRA_FOR_TRADE` and `EXTRA_FOR_QUERY` which are also accessed in that same activity but not further explored.

Shopping List Page



The activity for this screen is `ShoppingListActivity`, and its package is `pt.pingodoce.presentation.shoppingList`.

Shopping List (Object)

This object seems to describe a single Shopping List, which are created by the user in the above screen.

The data structure for shopping lists is the following:

```
description - string
isActive - boolean
products - List of ShoppingListProduct
sharedWith - List of Friend
dateCreated - String
dateUpdated - String
id - String
clientId - String
userId - String
name - String
```

This is an average Java Object with getters and setters and methods to, for example, add Friends to `sharedWith`.

Shopping List Category (Object)

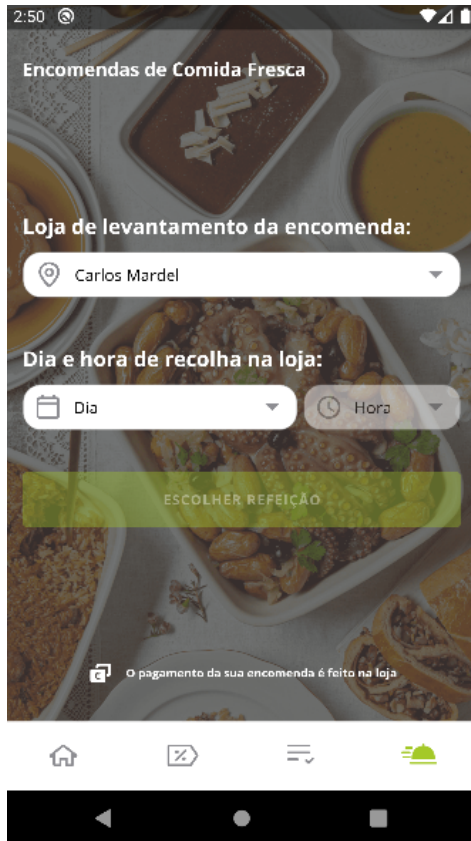
```
name - string
color - string
sortOrder - int
```

Shopping List Product (Object)

```
id - int
category - ShoppingListCategory
name - string
unitLabel - string
badgeUrl - string
badgeDisclaimer - string
showBadgeDisclaimer - bool
badgeDisclaimerTitle - string
promotionCode - string
priceLabel - string
```

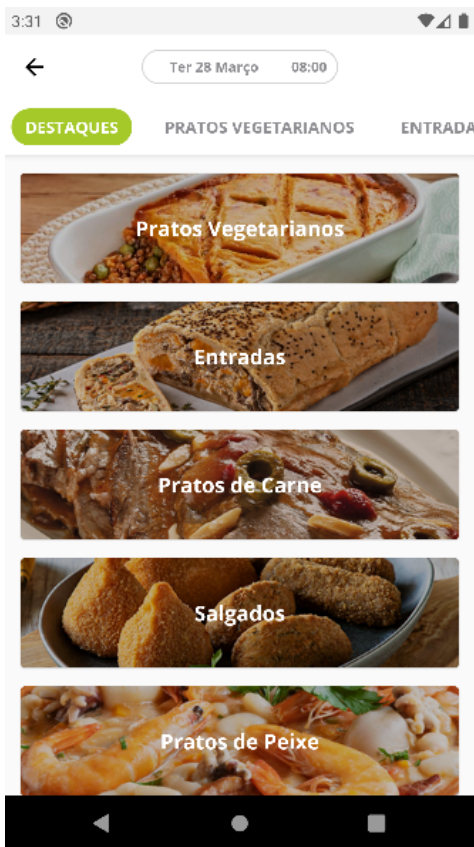
```
unitPriceLabel - string
dateUpdated - string
sortOrder - int
manualOrder - int
deleted - bool
timeStamp - long
quantity - int
checked - bool
```

TakeAway Page



The activity for this screen is `TakeAwayStoreActivity`, and its package is `pt.pingodoce.app.presentation.takeaway.pickupstore`

TakeAway Categories



The activity for this screen is `TakeAwayCategory`, and its package is `pt.pingodoce.app.data.local.models.takeaway`.

TakeAwayCategory (Data Model)

The class for this object is `TakeAwayCategory`, and its package is `pt.pingodoce.app.data.local.models.takeaway`.

Object structure:

```
code - string
name - string
imageUrl - string
isHighlight - boolean
```

API Interaction

This is the interface the app uses to interact with the API.

```
public interface TakeAwayApi {
    @POST("https://app-proxy.pingodoce.pt/api/v2/takeaway/orders/cancel")
    Object cancelOrder(@Body TakeAwayOrderCancellationRequest takeAwayOrderCancellationRequest, d<? super Response<u>> dVar);

    @GET("https://app-proxy.pingodoce.pt/api/v2/takeaway/categories")
    Object categories(@Query("storeId") int i10, @Query("pickUpDate") String str, d<? super Response<List<TakeAwayCategory>>> dVar);

    @GET("https://app-proxy.pingodoce.pt/api/v2/takeaway/categories/{id}")
    Object categoryProducts(@Path("id") String str, @Query("storeId") int i10, @Query("pickUpDate") String str2, @Query("pageNumber") int i11, @Query("pageSize") int i12, d<? super Response<List<TakeAwayProduct>>> dVar);

    @GET("https://app-proxy.pingodoce.pt/api/v2/takeaway/orders")
    Object orders(d<? super Response<List<TakeAwayOrder>>> dVar);

    @POST("https://app-proxy.pingodoce.pt/api/v2/takeaway/orders")
    Object placeOrder(@Body TakeAwayOrder takeAwayOrder, d<? super Response<TakeAwayOrderResponse>> dVar);

    @GET("https://app-proxy.pingodoce.pt/api/v2/takeaway/status")
    Object status(d<? super Response<TakeAwayStatus>> dVar);

    @GET("https://app-proxy.pingodoce.pt/api/v2/takeaway/store/{id}")
    Object takeAwaySchedules(@Path("id") int i10, d<? super Response<Store>> dVar);
}
```

These methods are called elsewhere in the app but the code involved was not understood. It is displayed below as-is:

The method `placeOrder` is invoked in `TakeAwayManager`:

```
/* compiled from: TakeAwayManager.kt */
@kotlin.coroutines.jvm.internal.f(
    c = "pt.pingodoce.app.data.managers.TakeAwayManager",
    f = "TakeAwayManager.kt",
    l = {179},
    m = "placeOrder")

public static final class h extends kotlin.coroutines.jvm.internal.d {
    /* synthetic */ Object f15724w;
    int f15726y;
    h(bd.d<? super h> dVar) {
        super(dVar);
        m1.this = r1;
    }
    @Override // kotlin.coroutines.jvm.internal.a
    public final Object invokeSuspend(Object obj) {
        this.f15724w = obj;
        this.f15726y |= Integer.MIN_VALUE;
        return m1.this.w(null, this);
    }
}
```

and in `TakeAwayService`:

```
@kotlin.coroutines.jvm.internal.f(
    c = "pt.pingodoce.app.data.remote.services.TakeAwayService$placeOrder$2",
    f = "TakeAwayService.kt",
    l = {44},
    m = "invokeSuspend"
)
/* loaded from: classes2.dex */
public static final class e extends kotlin.coroutines.jvm.internal.l
    implements id.l<bd.d<? super Response<TakeAwayOrderResponse>>, Object> {

    /* renamed from: x reason: collision with root package name */
    int f23868x;

    /* renamed from: z reason: collision with root package name */
    final /* synthetic */ TakeAwayOrder f23870z;

    /* JADX WARN: 'super' call moved to the top of the method (can break code semantics) */
    e(TakeAwayOrder takeAwayOrder, bd.d<? super e> dVar) {
        super(1, dVar);
        this.f23870z = takeAwayOrder;
    }

    @Override // kotlin.coroutines.jvm.internal.a
    public final bd.d<yc.u> create(bd.d<?> dVar) {
        return new e(this.f23870z, dVar);
    }

    @Override // id.l
    /* renamed from: d */
    public final Object invoke(bd.d<? super Response<TakeAwayOrderResponse>> dVar) {
        return ((e) create(dVar)).invokeSuspend(yc.u.f27252a);
    }

    @Override // kotlin.coroutines.jvm.internal.a
    public final Object invokeSuspend(Object obj) {
        Object d10;
        d10 = cd.d.d();
        int i10 = this.f23868x;
        if (i10 == 0) {
            yc.n.b(obj);
            TakeAwayApi takeAwayApi = i0.this.f23856d;
            TakeAwayOrder takeAwayOrder = this.f23870z;
            this.f23868x = 1;
            obj = takeAwayApi.placeOrder(takeAwayOrder, this);
            if (obj == d10) {

```

```

        return d10;
    }
    } else if (i10 != 1) {
        throw new IllegalStateException("call to 'resume' before 'invoke' with coroutine");
    } else {
        yc.n.b(obj);
    }
    return obj;
}
}
}

```

Confirm Order Page

3:32

X
A sua encomenda (1)

1
LX

Moussaka de Legumes com Crumble de Tomate

9.99€

Total
9.99€

DADOS DE LEVANTAMENTO

Carlos Mardel

Dia

Ter 28 Março

Hora

08:00

A encomenda ficará disponível para levantamento na data e hora indicada nestes campos.

OPS SEUS DADOS

Nome
....Com%20%3C<Script>Alert(1);</Script>

CONFIRMAR ENCOMENDA

E-mail
p3ngrey@gmail.com

Page layout

This function sets the title of the page (In the picture above, `A sua encomenda (1)`).

```

public final void X1(int i10) {
    String string;
    Toolbar toolbar = G1().f24771x;
    if (this.f18266h0) {
        c0 c0Var = c0.f13926a;
        String string2 = getString(R.string.lbl_take_away_checkout_alt);
        string = String.format(string2, Arrays.copyOf(new Object[]{Integer.valueOf(i10)}, 1));
    } else {
        string = getString(R.string.lbl_take_away_checkout);
    }
    toolbar.setTitle(string);
}
...

```

Orders View

![[Pasted image 20230324153440.png]]

It wasn't possible to ascertain for sure but the activity for this page is likely `TakeAwayOrderListActivity`, since looking for the string `Encomendas` in the app's resources points to `R.layout.activity_take_away_order_list`, which is returned here, likely to be inflated.

Dynamic Analysis

During the dynamic analysis of the `PingoDoce` app, we encountered some challenges due to SSL pinning implemented by the app. This prevented us from intercepting traffic using the certificate we installed on the testing emulator.

![[Pasted image 20230323172425.png]]

To overcome this issue, we turned to Frida, an open-source dynamic instrumentation toolkit, which allows us to bypass SSL Pinning.

After installing and setting up Frida, we used an SSL unpinning script to intercept the app's traffic using BurpSuite, this will be done in conjunction with the Root Check and VM Check bypass, the full script is present [here](https://gist.github.com/Pengrey/f573b2c3385f91327e8b3149bed364a3). The result is the following:

```
PS > powershell
PS > frida -U -l .\bypass.js -f pt.pingodoce

----
/ _ |   Frida 16.0.10 - A world-class dynamic instrumentation toolkit
| (| |
> _ |   Commands:
/_/ |_|   help      -> Displays the help system
. . . .   object?    -> Display information about 'object'
. . . .   exit/quit  -> Exit
. . . .
. . . .   More info at https://frida.re/docs/home/
. . . .
. . . .   Connected to Android Emulator 5554 (id=emulator-5554)
Spawning `pt.pingodoce`...
Spawned `pt.pingodoce`. Resuming main thread!

[Android Emulator 5554:pt.pingodoce ]->
=====
[#] Android Bypass for various Certificate Pinning methods [#]
=====
[-] OkHTTPv3 {1} pinner not found
[-] OkHTTPv3 {2} pinner not found
<SNIP>
[-] Apache AbstractVerifier check not found
[-] Chromium Cronet pinner not found
[-] Flutter HttpCertificatePinning pinner not found
[-] Flutter SslPinningPlugin pinner not found

=====
[#] Root Beer Check Bypass [#]
=====
[+] Loaded 9137 classes!
[+] loaded: -1
[+] ProcessManager hook not loaded
[+] KeyInfo hook not loaded

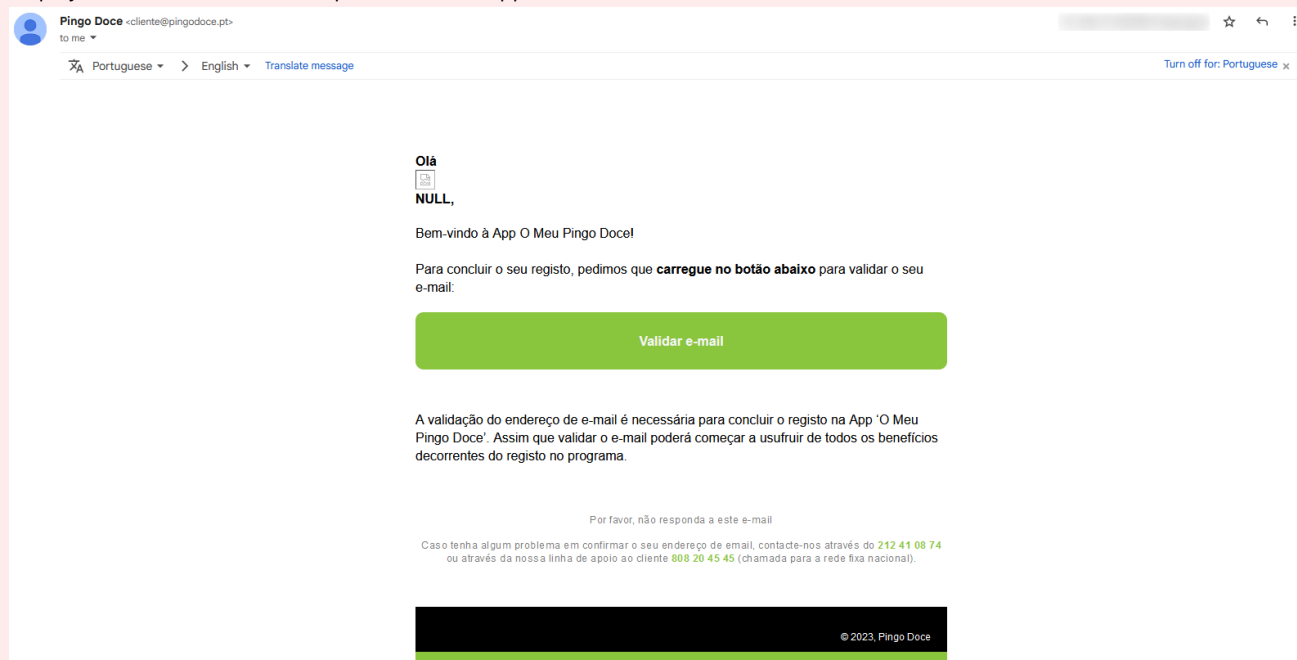
=====
[#] VM Check Bypass [#]
=====
[+] Bypass return value for binary: Superuser.apk
[+] Bypass return value for binary: Superuser.apk
[+] Bypass return value for binary: Superuser.apk
[+] Bypass return value for binary: Superuser.apk
<SNIP>
[!] Unexpected SSLPeerUnverifiedException occurred, trying to patch it dynamically...
[!] Attempting to bypass uncommon SSL Pinning method on: ff.g.b
```

After successfully bypassing the SSL Pinning implemented in the app, we were able to intercept and log all the requests made by the app to the server. The intercepted traffic can be observed in the graph generated by the following [URL](#).

⚡ Errors Found

During the dynamic analysis, we encountered two errors that were caused by the lack of sanitization of the username field in the application. The first error allowed the user to input valid HTML and JavaScript code in the username field, which could later be

displayed in emails or other components of the app.



However, we did not explore this bug any further.

The second error occurred when the lack of sanitization enabled the user to use characters that the backend was unable to handle. For example, when we used the character `%` in the username, it broke certain functions on the backend through the API. Request:

```
POST /api/v2/takeaway/orders HTTP/1.1
Host: app-proxy.pingodoce.pt
<SNIP>
Accept-Language: en-US
Content-Type: application/json; charset=UTF-8
Content-Length: 3333
Accept-Encoding: gzip, deflate
User-Agent: okhttp/4.10.0
Connection: close

{"number": "",
<SNIP>
"ownerName": "%",
<SNIP>
}
```

Response:

```
HTTP/1.1 400 Bad Request
Content-Length: 173
Connection: close
Content-Type: application/json
Date: Thu, 23 Mar 2023 18:45:26 GMT
api-supported-versions: 1,2
X-Powered-By: ASP.NET

{"errors": [{"status": "400", "code": "1024", "title": "Validation error: Content not allowed >", "detail": "'{0}' value is not valid for input", "source": {"parameter": "ownerName"}}]}
```

This issue could potentially lead to security vulnerabilities and needs to be addressed by implementing proper sanitization techniques.

After successfully bypassing the SSL Pinning implemented in the app, we were able to intercept and log all the requests made by the app to the server. The intercepted traffic can be observed in the graph generated by the following [URL](https://drive.google.com/file/d/1kPf7hn-EUVX1hdynTyHaNdIPy0yP-zQt/view?usp=sharing):

<https://drive.google.com/file/d/1kPf7hn-EUVX1hdynTyHaNdIPy0yP-zQt/view?usp=sharing>

This graph provides a view of the application's flow and functionalities.

Appendix - Manifests

CTT AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="95"
android:versionName="1.0.960" android:hardwareAccelerated="true" android:compileSdkVersion="33"
android:compileSdkVersionCodename="13" package="pt.ctt.outsystems.CTT" platformBuildVersionCode="33"
platformBuildVersionName="13">
    <uses-sdk android:minSdkVersion="26" android:targetSdkVersion="33"/>
    <supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true"
android:largeScreens="true" android:resizeable="true" android:xlargeScreens="true"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_MEDIA_AUDIO"/>
    <uses-permission android:name="android.permission.READ_MEDIA_IMAGES"/>
    <uses-permission android:name="android.permission.READ_MEDIA_VIDEO"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-feature android:name="android.hardware.location.gps"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <queries>
        <intent>
            <action android:name="android.media.action.IMAGE_CAPTURE"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.GET_CONTENT"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.PICK"/>
        </intent>
    </queries>
    <uses-permission android:name="com.amazon.device.messaging.permission.RECEIVE"/>
    <permission android:name="pt.ctt.outsystems.CTT.permission.RECEIVE_ADM_MESSAGE"
android:protectionLevel="signature"/>
    <uses-permission android:name="pt.ctt.outsystems.CTT.permission.RECEIVE_ADM_MESSAGE"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
    <queries>
        <intent>
            <action android:name="android.media.action.IMAGE_CAPTURE"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.GET_CONTENT"/>
        </intent>
    </queries>
    <permission android:name="pt.ctt.outsystems.CTT.permission.C2D_MESSAGE" android:protectionLevel="signature"/>
    <uses-permission android:name="pt.ctt.outsystems.CTT.permission.C2D_MESSAGE"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="com.sec.android.provider.badge.permission.READ"/>
    <uses-permission android:name="com.sec.android.provider.badge.permission.WRITE"/>
    <uses-permission android:name="com.htc.launcher.permission.READ_SETTINGS"/>
    <uses-permission android:name="com.htc.launcher.permission.UPDATE_SHORTCUT"/>
    <uses-permission android:name="com.sonyericsson.home.permission.BROADCAST_BADGE"/>
    <uses-permission android:name="com.sonymobile.home.permission.PROVIDER_INSERT_BADGE"/>
    <uses-permission android:name="com.anddoes.launcher.permission.UPDATE_COUNT"/>
    <uses-permission android:name="com.majeur.launcher.permission.UPDATE_BADGE"/>
    <uses-permission android:name="com.huawei.android.launcher.permission.CHANGE_BADGE"/>
    <uses-permission android:name="com.huawei.android.launcher.permission.READ_SETTINGS"/>
    <uses-permission android:name="com.huawei.android.launcher.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="android.permission.READ_APP_BADGE"/>
    <uses-permission android:name="com.oppo.launcher.permission.READ_SETTINGS"/>
    <uses-permission android:name="com.oppo.launcher.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="me.everything.badger.permission.BADGE_COUNT_READ"/>
    <uses-permission android:name="me.everything.badger.permission.BADGE_COUNT_WRITE"/>
    <uses-feature android:name="android.hardware.camera" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
</manifest>
```



```

<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-feature android:name="android.hardware.wifi" android:required="false"/>
<application android:label="@string/app_name" android:icon="@mipmap/ic_launcher"
android:hardwareAccelerated="true" android:supportsRtl="true" android:extractNativeLibs="false"
android:networkSecurityConfig="@xml/network_security_config"
android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:isSplitRequired="true">
    <activity android:theme="@style/Theme.App.SplashScreen" android:label="@string/activity_name"
android:name="pt.ctt.outsystems.CTT.MainActivity" android:exported="true" android:launchMode="singleTop"
android:screenOrientation="portrait"
android:configChanges="smallestScreenSize|screenSize|uiMode|screenLayout|orientation|keyboardHidden|keyboard|locale"
android:windowSoftInputMode="adjustResize">
        <intent-filter android:label="@string/launcher_name">
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="pt.ctt.outsystems.ctt"/>
        </intent-filter>
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="pt.ctt.outsystems.ctt" android:host="appserver.ctt.pt"/>
        </intent-filter>
    </activity>
    <activity android:theme="@style/zxing_CaptureTheme"
android:name="com.outsystems.plugins.barcodeScanner.CustomScannerActivity" android:exported="false"
android:stateNotNeeded="true" android:screenOrientation="fullSensor"
android:windowSoftInputMode="stateAlwaysHidden"/>
        <provider android:name="org.apache.cordova.camera.FileProvider" android:exported="false"
android:authorities="pt.ctt.outsystems.CTT.camera.provider" android:grantUriPermissions="true">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS"
android:resource="@xml/camera_provider_paths"/>
        </provider>
        <activity android:theme="@android:style/Theme.Black.NoTitleBar"
android:name="com.outsystems.imageeditor.view.ImageEditorActivity"/>
        <enable-feature android:name="com.amazon.device.messaging" android:required="false"/>
        <service android:name="com.onesignal.ADMMessageHandler" android:exported="true"/>
        <receiver android:name="com.onesignal.ADMMessageHandler.Receiver"
android:permission="com.amazon.device.messaging.permission.SEND" android:exported="true">
            <intent-filter>
                <action android:name="com.amazon.device.messaging.intent.REGISTRATION"/>
                <action android:name="com.amazon.device.messaging.intent.RECEIVE"/>
                <category android:name="pt.ctt.outsystems.CTT"/>
            </intent-filter>
        </receiver>
        <meta-data android:name="onesignal_app_id" android:value=""/>
        <meta-data android:name="onesignal_google_project_number" android:value="str:REMOTE"/>
        <receiver android:name="com.onesignal.GcmBroadcastReceiver"
android:permission="com.google.android.c2dm.permission.SEND" android:exported="true">
            <intent-filter android:priority="999">
                <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
                <category android:name="pt.ctt.outsystems.CTT"/>
            </intent-filter>
        </receiver>
        <receiver android:name="com.onesignal.NotificationOpenedReceiver" android:exported="true"/>
        <service android:name="com.onesignal.HmsMessageServiceOneSignal" android:exported="false">
            <intent-filter>
                <action android:name="com.huawei.push.action.MESSAGING_EVENT"/>
            </intent-filter>
        </service>
        <activity android:theme="@android:style/Theme.Translucent.NoTitleBar"
android:name="com.onesignal.NotificationOpenedActivityHMS" android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.VIEW"/>
            </intent-filter>
        </activity>
        <service android:name="com.onesignal.GcmIntentService" exported="false"/>
        <service android:name="com.onesignal.GcmIntentJobService"
android:permission="android.permission.BIND_JOB_SERVICE" exported="false"/>

```

```

        <service android:name="com.onesignal.RestoreJobService"
android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
        <service android:name="com.onesignal.RestoreKickoffJobService"
android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
        <service android:name="com.onesignal.SyncService" android:stopWithTask="true"/>
        <service android:name="com.onesignal.SyncJobService"
android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
        <activity android:theme="@android:style/Theme.Translucent.NoTitleBar"
android:name="com.onesignal.PermissionsActivity" android:exported="false"/>
        <service android:name="com.onesignal.NotificationRestoreService" android:exported="false"/>
        <receiver android:name="com.onesignal.BootUpReceiver" android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED"/>
                <action android:name="android.intent.action.QUICKBOOT_POWERON"/>
            </intent-filter>
        </receiver>
        <receiver android:name="com.onesignal.UpgradeReceiver" android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MY_PACKAGE_REPLACED"/>
            </intent-filter>
        </receiver>
        <service android:name="com.google.firebase.messaging.FirebaseMessagingService" android:exported="true">
            <intent-filter android:priority="-500">
                <action android:name="com.google.firebase.MESSAGING_EVENT"/>
            </intent-filter>
        </service>
        <service android:name="com.google.firebase.components.ComponentDiscoveryService" android:exported="false">
            <meta-data android:name="com.google.firebase.components:com.google.firebase.iid.Registrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        </service>
        <receiver android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver"
android:permission="com.google.android.c2dm.permission.SEND" android:exported="true">
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
            </intent-filter>
        </receiver>
        <service android:name="com.google.firebase.iid.FirebaseInstanceIdService" android:exported="true">
            <intent-filter android:priority="-500">
                <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
            </intent-filter>
        </service>
        <activity android:theme="@android:style/Theme.Translucent.NoTitleBar"
android:name="com.google.android.gms.common.api.GoogleApiActivity" android:exported="false"/>
        <provider android:name="com.google.firebase.provider.FirebaseInitProvider" android:exported="false"
android:authorities="pt.ctt.outsystems.CTT.firebaseinitprovider" android:initOrder="100"/>
        <meta-data android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version"/>
        <provider android:name="androidx.startup.InitializationProvider" android:exported="false"
android:authorities="pt.ctt.outsystems.CTT.androidx-startup">
            <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer" android:value="androidx.startup"/>
            <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer"
android:value="androidx.startup"/>
        </provider>
        <activity android:theme="@style/zxing_CaptureTheme"
android:name="com.journeyapps.barcodescanner.CaptureActivity" android:clearTaskOnLaunch="true"
android:stateNotNeeded="true" android:screenOrientation="sensorLandscape"
android:windowSoftInputMode="stateAlwaysHidden"/>
        <meta-data android:name="com.android.vending.splits.required" android:value="true"/>
        <meta-data android:name="com.android.stamp.source" android:value="https://play.google.com/store"/>
        <meta-data android:name="com.android.stamp.type" android:value="STAMP_TYPE_DISTRIBUTION_APK"/>
        <meta-data android:name="com.android.vending.splits" android:resource="@xml/splits0"/>
        <meta-data android:name="com.android.vending.derived.apk.id" android:value="3"/>
    </application>
</manifest>

```

PingoDoce AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="374"
android:versionName="3.1.5" android:installLocation="preferExternal" android:compileSdkVersion="32"
android:compileSdkVersionCodename="12" package="pt.pingodoce" platformBuildVersionCode="32"
platformBuildVersionName="12">
    <uses-sdk android:minSdkVersion="24" android:targetSdkVersion="32"/>

```

```

<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.USE_BIOMETRIC"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-feature android:name="android.hardware.location.gps" android:required="true"/>
<queries>
    <intent>
        <action android:name="android.speech.RecognitionService"/>
    </intent>
    <intent>
        <action android:name="android.intent.action.VIEW"/>
        <data android:scheme="https"/>
    </intent>
    <package android:name="com.google.android.apps.maps"/>
</queries>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.USE_FINGERPRINT"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-feature android:glEsVersion="0x20000" android:required="true"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
<uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-feature android:name="android.hardware.wifi" android:required="false"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<application android:theme="@style/App" android:label="@string/app_name_label"
    android:icon="@mipmap/ic_launcher" android:name="pt.pingodoce.app.PingoDoceApp" android:allowBackup="false"
    android:supportRtl="true" android:extractNativeLibs="false" android:roundIcon="@mipmap/ic_launcher_round"
    android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:isSplitRequired="true">
    <activity android:theme="@style/ZxingTheme" android:name="com.journeyapps.barcodescanner.CaptureActivity"
        android:clearTaskOnLaunch="true" android:stateNotNeeded="true" android:screenOrientation="portrait"
        android:windowSoftInputMode="stateAlwaysHidden"/>
    <activity android:theme="@style/App.Splash"
        android:name="pt.pingodoce.app.presentation.landing.splash.SplashActivity" android:exported="true"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activity android:name="pt.pingodoce.app.services.PushNotificationDispatcherActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.services.NotificationLandingActivity"
        android:clearTaskOnLaunch="true" android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.landing.walkthrough.AppWalkThroughActivity"
        android:screenOrientation="portrait"/>
    <activity android:theme="@style/App.Intro"
        android:name="pt.pingodoce.app.presentation.landing.intro.IntroActivity" android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.phone.validate.LoginPhoneActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.phone.update.UpdatePhoneActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.phone.update.CreatePhoneActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.sms.validate.SmsValidationActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.sms.update.PhoneUpdateSmsValidationActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.pin.CreatePinActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.pin.insert.InsertPinActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.phone.update.ValidatePinForUpdatePhoneActivity"
        android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.email.OnboardingEmailActivity"

```

```
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.nif.OnboardingNifActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.mypd.OnboardingMyPdActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.onboarding.email.emailsent.OnboardingEmailSentActivity"
android:exported="true" android:launchMode="singleTask" android:screenOrientation="portrait">
    <intent-filter android:label="@string/app_name_label">
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE"/>
        <data android:scheme="@string/deeplink_scheme_onboarding"
android:host="@string/deeplink_host_onboarding" android:pathPrefix="@string/deeplink_prefix_onboarding"/>
    </intent-filter>
    </activity>
    <activity android:name="pt.pingodoce.app.presentation.login.phone.terms.FetchTermsAndConditionsActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.login.phone.terms.TermsAndConditionsActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.success.OnboardingSuccessActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.userdata.insert.UserDataInsertActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.userdata.update.UserDataUpdateActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.onboarding.loyaltycard.OnboardingCardQuestionActivity"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.loyaltycard.OnboardingOwnerInv2Activity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.common.loyaltycard.virtualcard.VirtualCardDataConfirmationActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.common.loyaltycard.virtualcard.VirtualCardPreferencesActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.common.loyaltycard.virtualcard.VirtualCardSuccessActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.onboarding.loyaltycard.OnboardingOtherUserCardTutorialActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.onboarding.notifications.OnboardingNotificationActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.onboarding.loyaltycard.OnboardingCardTutorialActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.onboarding.loyaltycard.OnboardingCardRegistrationActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.home.loyaltycard.resume.AvailableCardsResumeActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.home.loyaltycard.resume.CardRegistrationResumeActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.home.loyaltycard.landing.CardRegistrationLandingActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.home.loyaltycard.registration.CardRegistrationActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.home.loyaltycard.status.CardRegistrationStatusActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.common.loyalty.removed.LoyaltyRemovedActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.clubs.ClubActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.clubs.baby.BabyClubUpdateActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.clubs.baby.BabyClubHomeActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.clubs.baby.BabyClubWelcomeActivity"
android:screenOrientation="portrait"/>
```

```
<activity android:name="pt.pingodoce.app.presentation.clubs.wine.WineClubWelcomeActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.clubs.wine.WineClubHomeActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.clubs.wine.WineClubUpdateActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.login.pin.recovery.PinRecoverActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.login.pin.change.RecoverPinActivity"
android:exported="true" android:launchMode="singleTask" android:screenOrientation="portrait">
    <intent-filter android:label="@string/app_name_label">
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE"/>
        <data android:scheme="@string/deeplink_scheme_pin" android:host="@string/deeplink_host"
android:pathPrefix="@string/deeplink_prefix_pin"/>
    </intent-filter>
</activity>
<activity android:name="pt.pingodoce.app.presentation.onboarding.resume.OnboardingResumeActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Home" android:name="pt.pingodoce.app.presentation.home.HomeActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.ShoppingList"
android:name="pt.pingodoce.app.presentation.shoppinglist.ShoppingListActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.ShoppingList.ManualSort"
android:name="pt.pingodoce.app.presentation.shoppinglist.manualsort.ManualSortShoppingListActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.common.loyaltycard.LoyaltyCardActivity"/>
<activity android:name="pt.pingodoce.app.presentation.common.versioncontrol.VersionControlActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.common.loyaltycard.requests.LoyaltyCardInvitesActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Search"
android:name="pt.pingodoce.app.presentation.common.search.SearchActivity" android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.common.product.ProductDetailActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.ProductList"
android:name="pt.pingodoce.app.presentation.promotion.PromotionActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/App.ProductList"
android:name="pt.pingodoce.app.presentation.home.tailored.TailoredForYouActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.ProductList"
android:name="pt.pingodoce.app.presentation.flyer.FlyerActivity" android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.shoppinglist.mylists.ShoppingListsActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Lock"
android:name="pt.pingodoce.app.presentation.landing.lock.LockActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Account"
android:name="pt.pingodoce.app.presentation.account.loggedAccount.LoggedAccountActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Account"
android:name="pt.pingodoce.app.presentation.account.anonymousAccount.AnonymousAccountActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Profile"
android:name="pt.pingodoce.app.presentation.account.loggedAccount.profile.ProfileActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.account.loggedAccount.profile.nif.EditNifActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.profile.birthdate.EditBirthDateActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.account.loggedAccount.profile.email.EditEmailActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.shoppinglist.shared.SharedWithActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.banner.BannerDetailActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Stores"
android:name="pt.pingodoce.app.presentation.home.defaultstore.DefaultStoreActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.home.defaultstore.detail.DefaultStoreDetailActivity"
android:screenOrientation="portrait"/>
```



```
<activity android:name="pt.pingodoce.app.presentation.purchases.list.PurchaseListActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.purchases.detail.PurchaseDetailActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.purchases.products.PurchaseProductsActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Settings"
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.SettingsActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Settings"
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.termsOfUse.TermsOfUseActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Settings"
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.notifications.NotificationsActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Settings"
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.madeforyou.MadeForYouActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.invoice.ElectronicInvoiceActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.invoice.phone.ElectronicInvoicePhoneActiv
ity" android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.invoice.email.ElectronicInvoiceEmailActiv
ity" android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.invoice.status.ElectronicInvoiceStatusAct
ivity" android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.invoice.email.ElectronicInvoiceEditEmaila
ctivity" android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.pin.ChangePinActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.pin.ReplacePinActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.settings.delete.DeleteAccountActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.takeaway.walkthrough.TakeAwayWalkThroughActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.TakeAway.Landing"
android:name="pt.pingodoce.app.presentation.takeaway.landing.anonymousSession.AnonymousSessionTakeAwayLandingActivit
y" android:screenOrientation="portrait"/>
<activity android:theme="@style/App.TakeAway.Landing"
android:name="pt.pingodoce.app.presentation.takeaway.landing.TakeAwayLandingActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.Stores"
android:name="pt.pingodoce.app.presentation.takeaway.pickupstore.TakeAwayStoreActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.takeaway.pickupstore.detail.TakeAwayStoreDetailActivity"
android:screenOrientation="portrait"/>
<activity android:theme="@style/App.TakeAway"
android:name="pt.pingodoce.app.presentation.takeaway.products.TakeAwayProductsActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.takeaway.products.detail.TakeAwayMenuDetailActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.takeaway.products.detail.TakeAwayMealDetailActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.takeaway.orders.checkout.TakeAwayCheckoutSuccessActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.takeaway.orders.list.TakeAwayOrderListActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.takeaway.orders.detail.TakeAwayOrderDetailActivity"
android:screenOrientation="portrait"/>
<activity
android:name="pt.pingodoce.app.presentation.takeaway.orders.checkout.email.TakeAwayEditEmailActivity"
android:screenOrientation="portrait"/>
<activity android:name="pt.pingodoce.app.presentation.benefits.BenefitsActivity"
android:screenOrientation="portrait"/>
<activity
```

```
android:name="pt.pingodoce.app.presentation.account.loggedAccount.loyalty.intro.LoyaltyIntroActivity"
android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.account.loggedAccount.loyalty.LoyaltyActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.loyalty.pending.LoyaltyPendingActivity"
android:screenOrientation="portrait"/>
    <activity
android:name="pt.pingodoce.app.presentation.account.loggedAccount.loyalty.associations.AssociationsActivity"
android:screenOrientation="portrait"/>
    <activity android:name="presentation.debug.DebugActivity" android:screenOrientation="portrait"/>
    <activity android:name="pt.pingodoce.app.presentation.debug.cache.CacheActivity"
android:screenOrientation="portrait"/>
    <meta-data android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version"/>
    <uses-library android:name="org.apache.http.legacy" android:required="false"/>
    <meta-data android:name="com.google.android.geo.API_KEY" android:value="@string/google_api_key"/>
    <service android:name="pt.pingodoce.app.services.PushNotificationMessagingService" android:exported="false">
        <intent-filter>
            <action android:name="com.google.firebase.MESSAGING_EVENT"/>
        </intent-filter>
    </service>
    <meta-data android:name="com.google.firebase.messaging.default_notification_channel_id"
android:value="@string/firebase_default_channel"/>
    <meta-data android:name="com.google.firebase.messaging.default_notification_icon"
android:resource="@drawable/ic_notification"/>
    <meta-data android:name="com.google.firebase.messaging.default_notification_color"
android:resource="@color/green"/>
    <meta-data android:name="firebase_crashlytics_collection_enabled" android:value="true"/>
    <meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/facebook_app_id"/>
    <provider android:name="module.core.utils.AppFileProvider" android:exported="false"
android:authorities="pt.pingodoce.files.provider" android:grantUriPermissions="true">
        <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/provider_paths"/>
    </provider>
    <provider android:name="com.facebook.internal.FacebookInitProvider" android:exported="false"
android:authorities="pt.pingodoce.FacebookInitProvider"/>
    <receiver android:name="com.facebook.CurrentAccessTokenExpirationBroadcastReceiver"
android:exported="false">
        <intent-filter>
            <action android:name="com.facebook.sdk.ACTION_CURRENT_ACCESS_TOKEN_CHANGED"/>
        </intent-filter>
    </receiver>
    <activity android:theme="@android:style/Theme.Translucent.NoTitleBar"
android:name="com.google.android.gms.auth.api.signin.internal.SignInHubActivity" android:exported="false"
android:excludeFromRecents="true"/>
    <service android:name="com.google.android.gms.auth.api.signin.RevocationBoundService"
android:permission="com.google.android.gms.auth.api.signin.permission.REVOCATION_NOTIFICATION"
android:exported="true" android:visibleToInstantApps="true"/>
    <service android:name="com.google.firebase.components.ComponentDiscoveryService" android:exported="false"
android:directBootAware="true">
        <meta-data
android:name="com.google.firebase.components:com.google.firebase.crashlytics.CrashlyticsRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data
android:name="com.google.firebase.components:com.google.firebase.messaging.FirebaseMessagingRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data
android:name="com.google.firebase.components:com.google.firebase.analytics.connector.internal.AnalyticsConnectorRegi
strar" android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data
android:name="com.google.firebase.components:com.google.firebase.remoteconfig.ktx.FirebaseRemoteConfigKtxRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data
android:name="com.google.firebase.components:com.google.firebase.remoteconfig.RemoteConfigRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data android:name="com.google.firebase.components:com.google.firebase.abt.component.AbtRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data
android:name="com.google.firebase.components:com.google.firebase.ktx.FirebaseCommonKtxRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data
android:name="com.google.firebase.components:com.google.firebase.installations.FirebaseInstallationsRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
        <meta-data
```

```

android:name="com.google.firebase.components:com.google.firebase.datatransport.TransportRegistrar"
android:value="com.google.firebase.components.ComponentRegistrar"/>
    </service>
    <receiver android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver"
android:permission="com.google.android.c2dm.permission.SEND" android:exported="true">
        <intent-filter>
            <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
        </intent-filter>
    </receiver>
    <service android:name="com.google.firebase.messaging.FirebaseMessagingService" android:exported="false"
android:directBootAware="true">
        <intent-filter android:priority="-500">
            <action android:name="com.google.firebase.MESSAGING_EVENT"/>
        </intent-filter>
    </service>
    <activity android:theme="@android:style/Theme.Translucent.NoTitleBar"
android:name="com.google.android.gms.common.api.GoogleApiActivity" android:exported="false"/>
    <receiver android:name="com.google.android.gms.measurement.AppMeasurementReceiver" android:enabled="true"
android:exported="false"/>
    <service android:name="com.google.android.gms.measurement.AppMeasurementService" android:enabled="true"
android:exported="false"/>
    <service android:name="com.google.android.gms.measurement.AppMeasurementJobService"
android:permission="android.permission.BIND_JOB_SERVICE" android:enabled="true" android:exported="false"/>
    <provider android:name="com.google.firebase.provider.FirebaseInitProvider" android:exported="false"
android:authorities="pt.pingodoce.firebaseinitprovider" android:initOrder="100" android:directBootAware="true"/>
    <service android:name="com.google.android.datatransport.runtime.backends.TransportBackendDiscovery"
android:exported="false">
        <meta-data android:name="backend:com.google.android.datatransport.cct.CctBackendFactory"
android:value="cct"/>
    </service>
    <service
android:name="com.google.android.datatransport.runtime.scheduling.jobscheduling.JobInfoSchedulerService"
android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
    <receiver
android:name="com.google.android.datatransport.runtime.scheduling.jobscheduling.AlarmManagerSchedulerBroadcastReceiv
er" android:exported="false"/>
        <provider android:name="androidx.lifecycle.ProcessLifecycleOwnerInitializer" android:exported="false"
android:multiprocess="true" android:authorities="pt.pingodoce.lifecycle-process"/>
        <provider android:name="androidx.startup.InitializationProvider" android:exported="false"
android:authorities="pt.pingodoce.androidx-startup">
            <meta-data android:name="androidx.profileinstaller.ProfileInstallerInitializer"
android:value="androidx.startup"/>
        </provider>
        <receiver android:name="androidx.profileinstaller.ProfileInstallReceiver"
android:permission="android.permission.DUMP" android:enabled="true" android:exported="true"
android:directBootAware="false">
            <intent-filter>
                <action android:name="androidx.profileinstaller.action.INSTALL_PROFILE"/>
            </intent-filter>
        </receiver>
        <activity android:name="com.google.android.play.core.missingsplits.PlayCoreMissingSplitsActivity"
android:enabled="false" android:exported="false" android:process=":playcore_missing_splits_activity"
android:stateNotNeeded="true" android:launchMode="singleInstance"/>
        <activity android:theme="@style/Theme.PlayCore.Transparent"
android:name="com.google.android.play.core.common.PlayCoreDialogWrapperActivity" android:enabled="false"
android:exported="false" android:stateNotNeeded="true"/>
        <service android:name="com.google.android.play.core.assetpacks.AssetPackExtractionService"
android:enabled="false" android:exported="true"/>
        <meta-data android:name="com.android.vending.splits.required" android:value="true"/>
        <meta-data android:name="com.android.stamp.source" android:value="https://play.google.com/store"/>
        <meta-data android:name="com.android.stamp.type" android:value="STAMP_TYPE_DISTRIBUTION_APK"/>
        <meta-data android:name="com.android.vending.splits" android:resource="@xml/splits0"/>
        <meta-data android:name="com.android.vending.derived.apk.id" android:value="4"/>
    </application>
</manifest>

```