

ALDEAMO Manual de Integración NOTIFICACIONES PUSH

www.aldeamo.com

MAN-PP-03-08 V1 09/03/2020

Contenido

1.	Control de Cambios	. 3
2.	Integración APP PUSH	.4
3.	Integración PUSH WEB	. 27

1. Control de Cambios

Versión	Fecha	Descripción de la Modificación	Responsable (s)
1	21/02/2020	Creación del manual de integración de NOTIFICACIONES PUSH	Luisa García

2. Integración PUSH APP

Integración Android

Guía rápida de integración del SDK de Android.

Tabla de contenidos

- ¿Qué necesitas para la integración?
- Integración
- <u>Añadiendo las dependencias del SDK</u>
- <u>Configuración de AndroidManifest.xml</u>
- <u>Servicios de Firebase</u>
- <u>Servicios de indigitall</u>
- Icono para las notificaciones
- Inicializar el SDK
- Validar la integración

¿Qué necesitas para la integración?

- Crear una cuenta de indigitall
- <u>Crear un proyecto</u> en tu cuenta de indigitall
- Necesitarás el **App Key** del proyecto. Lo puedes encontrar en la <u>consola de</u> <u>administración</u> dentro de la sección *Configuración* en la pestaña *Proyectos*
- Un <u>Server Key de Firebase</u>
- Android Studio
- Un dispositivo Android o emulador con los servicios de Google Play instalados para ejecutar la app

Integración

Este artículo muestra el desarrollo mínimo que hay que hacer para comenzar a registrar dispositivos y poder realizar las primeras campañas push.

Puedes verlo en ente vídeo tutorial o leer las instrucciones más abajo:

https://youtu.be/JUWQA6VFr0E

Añadiendo las dependencias del SDK

Lo primero que hay que hacer es abrir el fichero **app/build.gradle**. En la captura de pantalla se puede comprobar dónde encontrar este fichero *app/build.gradle*.

Atención: es el fichero *build.gradle* que se encuentra en la carpeta *app*, NO el de la raíz del proyecto.



La librería está disponible a través del repositorio <u>Maven Central</u>. *Maven* es una de las herramientas de gestión de librerías más usadas en *Android*. Para integrar el *SDK* de indigitall es necesario añadir las siguientes d3pendencias:

- La librería de soporte de Android
- La librería de mensajes de Firebase
- Los servicios de localización de Google Play Services
- El SDK de indigitall

```
android {
compileSdkVersion 28
defaultConfig {
minSdkVersion 16
targetSdkVersion 28
}
```

```
}
```

```
repositories {
mavenCentral()
}
```

dependencies {

implementation 'com.android.support:appcompat-v7:28.0.0' implementation 'com.google.firebase:firebase-messaging:18.0.0' implementation 'com.google.android.gms:play-services-location:16.0.0' implementation 'com.indigitall:android:3.1.+ '

Configuración de AndroidManifest.xml

En el fichero AndroidManifest.XML hay que añadir la configuración para:

- Los servicios de Firebase
- Los servicios del SDK de indigitall
- Establecer el icono de las notificaciones

Añadiendo los servicios de Firebase

Nuestro SDK necesita integrarse con tu proyecto de <u>FCM (Firebase Cloud Messaging)</u>. FCM realiza la conexión con el dispositivo para poder enviarle notificaciones push. Esta conexión se establece con el *Push Token*, un token efímero, único y generado por Google para cada dispositivo.

<service android:name="com.indigitall.android.services.FirebaseMessagingService">
<intent-filter>
<action android:name="com.google.firebase.MESSAGING_EVENT" />
</intent-filter>
</service>
<service android:name="com.indigitall.android.services.FirebaseInstanceIdService">
<intent-filter>
</action android:name="com.indigitall.android.services.FirebaseInstanceIdService">
</action android

Añadiendo los servicios de indigitall

Estos servicios son necesarios para que nuestro SDK pueda sincronizar los datos del dispositivo con los servidores de indigitall.

<!-- OBLIGATORIOS -->

<!-- Para que cuando el usuario pulse una push, se guarde la métrica -->
<service android:name="com.indigitall.android.services.StatisticService">

<!-- Sincronización diaria de los datos del dispositivo -->
<service android:name="com.indigitall.android.services.NightService">

<!-- Para iniciar los servicios cuando reinicias el dispositivo -->

<receiver android:name="com.indigitall.android.receivers.BootReceiver"> <intent-filter>

<action android:name="android.intent.action.BOOT_COMPLETED" /> </intent-filter>

</receiver>

<!-- OPCIONALES -->

<!-- Para que cuando el usuario pulse un mensaje InApp, se guarde la métrica. Solo es necesario si usas la funcionalidad de mensajes InApp --> <service android:name="com.indigitall.inapp.services.StatisticInAppService" />

<!-- Para obtener la localización del dispositivo. Solo es necesario si vas a pedir permiso de localización para segmentar las pushes por la ubicación del dispositivo --> <receiver android:name="com.indigitall.android.receivers.LocationReceiver"> <intent-filter> <intent-filter> </action android:name="LocationReceiver.Action.LOCATION_UPDATE" /> </intent-filter> </receiver> </receiver>

Estableciendo el icono de las notificaciones

Este icono se mostrará en la barra superior del sistema Android y en la cabecera de las pushes enviadas a través de tu app.

Debe ser un icono monocromo, es decir, que la imagen debe contener solo un color y alfa.

<!-- Recurso para el icono monocromo -->
<meta-data android:name="indigitall.icon"
android:resource="@drawable/ic_slept_very_good_unselected"/>

<!-- Recurso para el color del icono --> <meta-data android:name="indigitall.color" android:resource="@color/colorPrimary"/>

Inicializar el SDK

Para inicializar el SDK es necesario **llamar al método init**. Esta llamada ha de producirse dentro del objeto *Application*.

La clase *Application* del SDK de Android es la clase base contenedora, y es lo primero que se ejecuta al abrir la aplicación. Por lo tanto, es necesario que nuestra aplicación tenga una clase personalizada que extienda de la clase *Application*.

Dentro de esta clase, en el método onCreate, debemos añadir las siguientes líneas de código:

Indigitall.init(this, "<your_indigitall_app_key>", "<your_firebase_sender_id")</pre>

- **your_indigitall_app_key** es una cadena alfanumérica que identifica tu proyecto de indigitall. Se obtiene desde la <u>consola de indigitall</u>
- **your_firebase_sender_id** es una cadena numérica que identifica tu proyecto de Firebase. Se obtiene desde la <u>consola de Firebase</u>

No olvides añadir el nombre de tu clase Application en el manifest AndroidManifest.xml:

<application android:name=".application.App">

Validar la integración

Para comprobar que la integración se ha realizado correctamente realiza lo siguiente:

1. Desde Android Studio, ves a la **pestaña Logcat** y busca la llamada PUT /device conteniendo los parámetros *appKey*, *deviceId* y *pushToken* y que devuelva *HTTP 200*.

https://youtu.be/kUqXNsN8Egc

2. Envía una notificación desde la consola de indigitall. Es posible que en la consola el contador de dispositivos aparezca a 0. No te preocupes, puede tardar unos minutos en actualizarse, pero no hace falta que esperes, la push debería llegar igualmente.

https://youtu.be/2BekYNAbwsg

Integración iOS

Guía rápida de integración del SDK de iOS.

Tabla de contenidos

<u>¿Qué necesitas para la integración?</u> <u>Integración</u> <u>Añadiendo las dependencias del SDK</u> <u>Implementación en AppDelegate</u> <u>Notification Service Extension</u> <u>Inicializar el SDK</u> <u>Validar la integración</u>

¿Qué necesitas para la instalación?

<u>Crear una cuenta</u> de indigitall <u>Crear un proyecto</u> en tu cuenta de indigitall Necesitarás el App Key del proyecto. Lo puedes encontrar en la <u>consola de</u> <u>administración</u> dentro de la sección Configuración en la pestaña Proyectos Un certificado válido de push para iOS. Aprende como <u>obtener el certificado push desde</u> <u>APNS</u> <u>Xcode</u>

Un dispositivo con iOS para ejecutar la app

Integración

Este artículo muestra el desarrollo mínimo que hay que hacer para comenzar a registrar dispositivos y poder realizar las primeras campañas push.

Añadiendo las dependencias del SDK

El SDK está disponible a través de <u>CocoaPods</u>.

CocoaPods es un gestor de dependencias válido para Swift y Objective-C, siendo el más popular en el desarrollo de iOS.

https://youtu.be/DhBuylp0fsl

Si aún no lo tienes, instala CocoaPods. Abre tu terminal y ejecuta los siguientes comandos:

\$ cd /ROOT/OF/YOUR/PROJECT \$ gem install cocoapods \$ pod init

Modifica el fichero PodFile de tu proyecto y añade este código:

target '<YourTarget>' do
 pod 'indigitall-ios'
end
target '<YourTargetNotificationExtension>' do
 pod 'indigitall-ios'
end
Recuerda: añade los pod correspondientes del SDK dentro de los nombres de los target que
tenga tu aplicación.

Atención: a partir de aquí debes usar .workspace en vez de .xcproject para trabajar en el

Actualiza el repositorio de CocoaPod e instala las dependencias desde el terminal:

proyecto. La principal diferencia es que .xcproject es para un único proyecto y .workspace puede contener varios proyectos. Activa las capabilities: Push Notifications en Background Modes Location updates **Background Fetch Remote notifications Push Notifications** ON Steps: ✓ Add the Push Notifications feature to your App ID. ✓ Add the Push Notifications entitlement to your entitlements file ▼ U Background Modes Modes: Audio, AirPlay, and Picture in Picture Location updates Newsstand downloads External accessory communication Uses Bluetooth LE accessories Acts as a Bluetooth LE accessory Background fetch Remote notifications Steps: ✓ Add the Required Background Modes key to your info plist file

\$ pod repo update

\$ pod install

Implementación en AppDelegate

Puedes verlo en nuestro vídeo tutorial o leer las instrucciones más abajo:

https://youtu.be/qmhzsSb9VXA

Mofifica el fichero AppDelegate para importar el SDK y sobreescribir los siguientes métodos:

Swift Objective C import Indigital

```
func application(_ application: UIApplication,
didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {
    Indigitall.set(deviceToken)
}
```

func application(_ application: UIApplication, didReceiveRemoteNotification data: [AnyHashable : Any]) {

Indigitall.handle(notification: data)

}

func application(_ application: UIApplication, handleActionWithIdentifier identifier: String?, forRemoteNotification userInfo: [AnyHashable : Any], completionHandler: @escaping () -> Void) {

Indigitall.handle(notification: userInfo, identifier: identifier)

}

```
func application(_ application: UIApplication, didReceiveRemoteNotification userInfo:
[AnyHashable : Any], fetchCompletionHandler completionHandler: @escaping
(UIBackgroundFetchResult) -> Void) {
```

Indigitall.handle(notification: userInfo)

```
completionHandler(.newData)
```

}

func application(_ application: UIApplication, performFetchWithCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {

Indigitall.performFetchWithCompletionHandler(completionHandler: completionHandler)
}

Notification Service Extension

Desde la salida de iOS 10, las apps pueden gestionar notificaciones push enriquecidas, es decir, con imagen, gif, vídeo, botones, etc.

Para poder hacer uso de estas funcionalidades, tu app necesita implementar el Notification Service Extensión.

Añade un nuevo Notification Service extensión a tu proyecto (Xcode: Archivo > Nuevo > Target).

Añade la extensión target en tu aplicación.

Crea un nuevo fichero llamado NotificationService dentro de este target. Sobrescribe todo el contenido con el siguiente código:

Swift Objective C

import Indigitall
class NotificationService: INNotificationServiceExtension {}

Inicializar el SDK

Añade el siguiente código al AppDelegate y recuerda cambiar YOUR-APP-KEY por el App Key que puedes encontrar en la <u>consola de indigitall</u>

Swift Objective C

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

Indigitall.initialize(appKey: "<YOUR-APP-KEY>") return true

}

Validar la integración

Para comprobar que la integración se ha realizado correctamente realiza lo siguiente:

Desde xCode, ves al log y busca la llamada PUT /device conteniendo los parámetros appKey, deviceId y pushToken y que devuelva HTTP 200.

https://youtu.be/kUqXNsN8Egc

Envía una notificación desde la consola de indigitall. Es posible que en la consola el contador de dispositivos aparezca a 0. No te preocupes, puede tardar unos minutos en actualizarse, pero no hace falta que esperes, la push debería llegar igualmente.

https://youtu.be/2BekYNAbwsg

Integración Cordova

Guía rápida de integración del SDK de Cordova.

Tabla de contenidos

- ¿Qué necesitas para la integración?
- Integración
- Importar el plugin
- <u>Configuración para Android</u>
- <u>Configuración para iOS</u>
- Inicialización del SDK
- Validar la integración

¿Qué necesitas para la integración?

- <u>Crear una cuenta</u> de indigitall
- <u>Crear un proyecto</u> en tu cuenta de indigitall
- Necesitarás el **App Key** del proyecto. Lo puedes encontrar en la <u>consola de</u> <u>administración</u> dentro de la sección *Configuración* en la pestaña *Proyectos*
- Para Android
- Un Server Key de Firebase
- <u>Android Studio</u>
- Un dispositivo Android o emulador con los servicios de Google Play instalados para ejecutar la app
- Para iOS
- Un certificado válido de push para iOS. Aprende como <u>obtener el certificado push</u> desde APNS
- Xcode
- Un dispositivo con iOS para ejecutar la app

Integración

Este artículo muestra el desarrollo mínimo que hay que hacer para comenzar a registrar dispositivos y poder realizar las primeras campañas push.

Puedes verlo en ente vídeo tutorial o leer las instrucciones más abajo:

https://youtu.be/7jg0kXo6qik

Nuestro SDK está disponible a través de <u>npm</u>.

npm (Node Package Manager) es un sistema de gestión de paquetes. Consiste en un cliente de línea de comandos y una base de datos en línea de paquetes públicos y privados.

Importar el plugin

Para importar el SDK a tu proyecto realiza estos pasos:

1. Abre la consola y posiciónate en la raíz del proyecto.

\$ cd /PATH/TO/YOUR/PROJECT

2. Ejecuta esta línea en la consola para importar el plugin:

\$ cordova plugin add indigitall-cordova-plugin

Configuración para Android

Si no quieres compilar tu proyecto Cordova para dispositivos Android, puedes saltarte este punto.

En caso contrario realiza estos pasos:

- 1. Para comenzar necesitas un fichero llamado **google-services.json**. Este fichero lo podrás exportar desde la <u>consola de Firebase</u>.
- 2. Muevelo a la carpeta raíz de tu proyecto. Deberías tener algo parecido a esto:



3. Añade el siguiente código en el fichero config.xml dentro de la sección android.

<platform name="android"> <resource-file src="google-services.json" target="/google-services.json" /> </platform>

Configuración para iOS

Desde la salida de **iOS 10**, las apps pueden gestionar **notificaciones push enriquecidas**, es decir, con imágen, gif, vídeo, botones, etc.

Para poder hacer uso de estas funcionalidades, tu app necesita implementar el *Notification Service Extension*.

- 1. Añade un nuevo Notification Service Extension a tu proyecto (Xcode: Archivo > Nuevo > Target).
- 2. Añade el extension target en tu aplicación.
- 3. **Crea un nuevo fichero** llamado *NotificationService* dentro de este *target*. Sobrescribe todo el contenido con el siguiente código:

Swift Objective C

import Indigitall
class NotificationService: INNotificationServiceExtension {}

Inicialización del SDK

Inicializa el SDK con el **método indigitall.init()** para comenzar a recibir notificaciones push. Esta inicialización debe realizarse **dentro del index.html** de tu proyecto.

window.plugins.indigitall.init({
 appKey: "<YOUR_APP_KEY>",
 senderId: "<YOUR_SENDER_ID>"
});

- YOUR_APP_KEY es una cadena de texto *alfanumérica*. La puedes encontrar en la <u>consola</u> <u>de administración</u> dentro de la sección *Configuración* en la pestaña *Proyectos*
- **YOUR_SENDER_ID** es una cadena *numérica*. Lo puedes encontrar en la <u>consola de</u> <u>Firebase</u>, dentro del proyecto, en la configuración, pestaña *Mensajería en la nube*.



indigitall v2	•		
Conf	iguración		
General	Mensajería en la nube	Integraciones	
	Credenciales de proyect	to	
	Clave	Toke AAA	
~	Clave de servidor	Kilcs 12Bł	
	Clave de servidor here	eda Alza	
	764695417263		

Validar la integración

Para comprobar que la integración se ha realizado correctamente realiza lo siguiente:

1. Desde Android Studio, ves a la **pestaña Logcat** y busca la llamada PUT /device conteniendo los parámetros *appKey*, *deviceId* y *pushToken* y que devuelva *HTTP 200*.

https://youtu.be/kUqXNsN8Egc

2. **Envía una notificación** desde la consola de indigitall. Es posible que en la consola el contador de dispositivos aparezca a 0. No te preocupes, puede tardar unos minutos en actualizarse, pero no hace falta que esperes, la push debería llegar igualmente.

https://youtu.be/2BekYNAbwsg

Integración Xamarin Guía rápida de integración del SDK de Xamarin.

Tabla de contenidos

- ¿Qué necesitas para la integración?
- Integración
- Importar el plugin
- <u>Configuración para Android</u>
- <u>Configuración para iOS</u>
- Notification Service Extension
- Inicialización del SDK
- Inicialización avanzada del SDK
- Get Device
- <u>Activar las notificaciones geolocalizadas</u>
- Configuración localización para Android
- <u>Configuración localización para iOS</u>
- Opciones que debes tener en cuenta
- Validar la integración

¿Qué necesitas para la integración?

- <u>Crear una cuenta</u> de indigitall
- <u>Crear un proyecto</u> en tu cuenta de indigitall
- Necesitarás el **App Key** del proyecto. Lo puedes encontrar en la <u>consola de</u> <u>administración</u> dentro de la sección *Configuración* en la pestaña *Proyectos*
- Para Android
- Un <u>Server Key de Firebase</u>
- <u>Android Studio</u>
- Un dispositivo Android o emulador con los servicios de Google Play instalados para ejecutar la app
- Para iOS
- Un certificado válido de push para iOS. Aprende como <u>obtener el certificado push</u> <u>desde APNS</u>
- <u>Xcode</u>
- Un dispositivo con iOS para ejecutar la app

Integración

Este artículo muestra el desarrollo mínimo que hay que hacer para comenzar a registrar dispositivos y poder realizar las primeras campañas push.

Nuestro SDK está disponible a través de <u>NuGet</u>.

NuGet es un sistema de gestión de paquetes. Consiste en un cliente de línea de comandos y una base de datos en línea de paquetes públicos y privados.

Importar el plugin

Para añadir nuestro SDK a tu proyecto a través de **NuGet** tienes que buscar el **paquete Com.Indigitall.Xamarin**.

Añade este paquete a tu proyecto (PCL, Android e iOS) de la siguiente manera:

https://youtu.be/WTQTV8NIJEg

Esta integración se ha realizado con el IDE Visual Studio.

Configuración para Android

Puedes verlo en ente vídeo tutorial o leer las instrucciones más abajo:

https://youtu.be/JUWQA6VFr0E

1. Para comenzar necesitas un fichero llamado **google-services.json**. Este fichero lo podrás exportar desde la <u>consola de Firebase</u>. Muevelo a la carpeta **raíz de tu proyecto**.

2. Añade los servicios de indigitall en tu AndroidManifest.xml como puedes ver más abajo.

<!-- START indigitall permissions -->

<uses-permission android:name="android.permission.INTERNET" /> <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" /> <uses-permission android:name="android.permission.VIBRATE" /> <uses-permission android:name="android.permission.WAKE_LOCK" /> <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" /> <!-- END indigitall permissions -->

<application>

```
<!-- START indigitall services -->
<service android:name="com.indigitall.android.services.StatisticService" />
<service android:name="com.indigitall.android.services.NightService" />
<receiver android:name="com.indigitall.android.receivers.BootReceiver">
<ireceiver android:name="com.indigitall.android.services.NightService" />
<receiver android:name="com.indigitall.android.receivers.BootReceiver">
<intent-filter>
</intent-filter>
</intent-filter>
<//receiver><//receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver></receiver>
```

<receiver android:name="com.indigitall.android.receivers.LocationReceiver">

<intent-filter>

```
<action android:name="LocationReceiver.Action.LOCATION_UPDATE" />
```

</intent-filter>

```
</receiver>
```

<service android:name="com.indigitall.android.services.FirebaseMessagingService">
 <intent-filter>

```
<action android:name="com.google.firebase.MESSAGING_EVENT" />
```

</intent-filter>

</service>

<meta-data android:name="indigitall.color" android:resource="@color/colorprimary" /> <meta-data android:name="indigitall.icon"

```
android:resource="@mipmap/launcher_foreground" />
```

```
<!-- END indigitall services -->
```

</application>

3. Añade el siguiente fragmento de código en el método *OnCreate* desde la que sea tu pantalla principal de la aplicación. Aquí la vamos a llamar *MyMainActivity*. El fragmento de código debe añadirse antes de llamar al método *LoadApplication* propio de *Xamarin*.

protected override void OnCreate(Bundle bundle)

```
DependencyService.Register<Com.Indigitall.Xamarin.Android.Indigitall>();
Com.Indigitall.Android.Indigitall.SetDefaultActivity(this, "YOUR ACTIVITY");
var app = new App();
Com.Indigitall.Xamarin.Android.Utils.PermissionUtils.RequestLocationPermission(this);
LoadApplication(app);
```

Configuración para iOS

Puedes verlo en ente vídeo tutorial o leer las instrucciones más abajo:

https://youtu.be/BoQ8QivsuNU

Para comenzar con la configuración de iOS, asegúrate tener OK estos puntos:

- Importa el plugin mediante <u>NuGet</u> como se menciona en la sección <u>importar el plugin</u>
- Establecer el Notification Center Delegate como se explica en la guía oficial

Nota: para que el SDK funcione correctamente el método *FinishedLaunching* debe ser llamado desde la clase *AppDelegate*, antes de que se produzca la llamada al método *LoadApplication*

La clase AppDelegate debería quedar así:

```
public override bool FinishedLaunching(UIApplication app, NSDictionary options)
  Forms.Init();
  DependencyService.Register<Com.Indigitall.Xamarin.iOS.Indigitall>();
 if (UIDevice.CurrentDevice.CheckSystemVersion(10, 0))
 {
    // iOS 10 or later
    var authOptions = UNAuthorizationOptions.Alert | UNAuthorizationOptions.Badge |
UNAuthorizationOptions.Sound;
    UNUserNotificationCenter.Current.RequestAuthorization(authOptions, (granted, error)
=>
    {
      if (granted)
        InvokeOnMainThread(() =>
        {
          UIApplication.SharedApplication.RegisterForRemoteNotifications();
        });
      }
    });
    // For iOS 10 display notification (sent via APNS)
    UNUserNotificationCenter.Current.Delegate = new
Com.Indigitall.Xamarin.iOS.UserNotificationCenterDelegate();
 }
 else
 {
   // iOS 9 or before
    var allNotificationTypes = UIUserNotificationType.Alert | UIUserNotificationType.Badge |
UIUserNotificationType.Sound;
    var settings = UIUserNotificationSettings.GetSettingsForTypes(allNotificationTypes, null);
    UIApplication.SharedApplication.RegisterUserNotificationSettings(settings);
 }
  UNUserNotificationCenter.Current.Delegate = new
Com.Indigitall.Xamarin.iOS.UserNotificationCenterDelegate();
  ....
  LoadApplication(new App());
```

return base.FinishedLaunching(app, options);
}
[Export("application:didRegisterForRemoteNotificationsWithDeviceToken:")]
override public void RegisteredForRemoteNotifications (UIApplication application, NSData
deviceToken)
IndigitaliXamarin.Indigitali.SetDeviceToken(deviceToken, (device) =>
{
U.
1
[Export("application:didReceiveRemoteNotification:fetchCompletionHandler:")]
override public void DidReceiveRemoteNotification (UIApplication application, NSDictionary
userInfo, Action <uibackgroundfetchresult> completionHandler)</uibackgroundfetchresult>
{
IndigitallXamarin.Indigitall.HandleWithNotification(userInfo, null);
}
[Export("application:handleActionWithIdentifier:forRemoteNotification:withResponseInfo:co
mpletionHandler:")]
override public void HandleAction (UIApplication application, string actionIdentifier,
NSDictionary remoteNotificationInfo, NSDictionary responseInfo, Action completionHandler)
{
IndigitalizationIndigitali.HandleWithNotification(remoteNotificationinio,
l

Notification Service Extension

Desde la salida de **iOS 10**, las apps pueden gestionar **notificaciones push enriquecidas**, es decir, con imágen, gif, vídeo, botones, etc.

Para poder hacer uso de estas funcionalidades, tu app necesita implementar el *Notification Service Extension*.

- 1. Añade un nuevo proyecto Notification Service Extension a tu solución
- 2. Añade la dependencia Com.Indigitall.Xamarin.NSE de NuGet
- 3. Referencia la extensión objetivo en tu proyecto *iOS*
- 4. Una vez que hayas creado la extensión, se crea un nuevo fichero dentro del proyecto. Es
- el NotificationService. Reemplaza su contenido por las siguientes líneas:

```
using System;
using Foundation;
using Com.Indigitall.Xamarin.iOS;
namespace Notification
{
    [Register("NotificationService")]
    public class NotificationService : NotificationServiceExtension
    {
        protected NotificationService(IntPtr handle) : base(handle) {}
    }
}
```

Inicialización del SDK

Puedes verlo en ente vídeo tutorial o leer las instrucciones más abajo:

https://youtu.be/kBpTAuNxi4M

Inicializa el *SDK* con la llamada al **metodo Indigital.init**. Primero hay que llamar a la interfaz *IIndigitall* dentro de tu clase *MainPage*, como vemos a continuación:

```
IIndigitall indigitall = DependencyService.Get<IIndigitall>();
if (indigitall != null)
{
    indigitall.Init("<YOUR_APP_KEY>", "<YOUR_SENDER_ID>");
}
```

- YOUR_APP_KEY es una cadena de texto *alfanumérica*. La puedes encontrar en la <u>consola</u> de administración dentro de la sección *Configuración* en la pestaña *Proyectos*
- **YOUR_SENDER_ID** es una cadena *numérica*. Lo puedes encontrar en la <u>consola de</u> <u>Firebase</u>, dentro del proyecto, en la configuración, pestaña *Mensajería en la nube*.



indigitall v2	indigitall v2 🔻						
Conf	Configuración						
General	Mensajería en la nube	Integraciones					
	Credenciales de proyect	to					
	Clave	Toke					
	Clave de servidor	AAA Kiles 12Bł					
	Clave de servidor here	eda Alza					
	ID del remitente ⑦						
	764695417263						

Inicialización avanzada del SDK

Nuestro SDK ofrece diversos callbacks que te ayudan tener mayor control del flujo de ejecución y a implementar comportamientos personalizados.

```
var indigitall = DependencyService.Get<IIndigitall>();
if (indigitall != null)
{
    indigitall.Init("YOUR APPKEY", "YOUR SENDER ID", true, (permissions, device)=>
    {
        //Cada vez que inicializa la sdk. AQui podéis coger el deviceID
        Console.WriteLine("device: " + device.deviceId);
    },(device)=>
    {
        Console.WriteLine("Newdevice: " + device.deviceId);
    },(errorCode,errorMessage)=>
    {
        Console.WriteLine("errorCode: " + errorCode+" Message: "+ errorMessage);
    });
```

Get Device

Puedes usar el método getDevice para obtener la información que ha registrado el SDK en referencia al dispositivo.

El callback de este método recibirá como parámetro el objeto device que contiene toda la información asociada al dispositivo.



Activar las notificaciones geolocalizadas

El *SDK* de indigitall puede gestionar la localización del usuario. Esto te permite usar los filtros de localización en la pantalla de enviar campaña push (<u>Campañas>Push>Nueva campaña</u> <u>push</u>>Filtros>Filtros geográficos)

	Inicio		🕅 🏺 🗰 🔍	Filtros geográficos 👔 ^	
D	Campañas	~		Áreas Dispositivos en las áreas	
*	Herramientas	~	appDermis - Ahora ;Bienvenidos! Hola desde indigitall		
Q	WhatsApp	~		Geolocalización / Regionales	
.h	Estadísticas	~		País	

Una vez hayamos habilitado esta funcionalidad, el usuario final tendrá que dar su consentimiento al permiso de localización y habilitar los servicios de localización de su smartphone, para que la aplicación obtenga la ubicación exacta del usuario.

Configuración localización para Android

Añade los permisos de localización incluyendo esta línea en el archivo AndroidManifest.xml:

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

Añade el siguiente código en el onCreate del MainActivity.cs y el método con la llamada OnRequestPermissionsResult para registrar la localización.

```
{
    DependencyService.Register<Com.Indigitall.Xamarin.Android.Indigitall>();
    //Com.Indigitall.Android.Indigitall.SetDefaultActivity(this,
    Java.Lang.Class.FromType(typeof(MyMainActivity)).Name);
    Com.Indigitall.Android.Indigitall.SetDefaultActivity(this, "YOUR ACTIVITY");
    ...
    var app = new App();
    ...
    Com.Indigitall.Xamarin.Android.Utils.PermissionUtils.RequestLocationPermission(this);
    ...
    LoadApplication(app);
    ...
  }
  public override void OnRequestPermissionsResult(int requestCode, string[] permissions,
    [GeneratedEnum] Permission[] grantResults)
    {
        base.OnRequestPermissionsResult(requestCode, permissions, grantResults);
        Com.Indigitall.Xamarin.Android.Utils.PermissionUtils.OnRequestPermissionResult(this,
        requestCode, permissions, grantResults);
    }
}
```

Configuración localización para iOS

Añade las siguientes claves en el archivo Info.plist de la aplicación.

```
<key>NSLocationAlwaysAndWhenInUseUsageDescription</key><string>Can we always use your location?</string>
```

<key>NSLocationAlwaysUsageDescription</key> <string>Can we always use your location?</string>

<key>NSLocationWhenInUseUsageDescription</key> <string>Can we use your location when using the apps?</string>

<key>UIBackgroundModes</key>

<array>

<string>location</string>

<string>remote-notification</string>

</array>

Lasclaves NSLocationAlwaysUsageDescription, NSLocationWhenInUseUsageDescription y NSL ocationAlwaysAndWhenInUseUsageDescription se pueden personalizar editando el contenido de la etiqueta < string >.

Opciones que debes tener en cuenta

Hay dos modos de gestionar los permisos de localización:

- Manual: Es la opción por defecto y es el desarrollador el encargado de gestionar los permisos de localización.
- Automática: Es el SDK el que gestiona los permisos de localización.

A continuación, te explicamos cómo configurar los permisos de localización en modo automático.

Hay que añadir el parámetro **true** cuando se inicialice el *SDK*. Esto se hace mediante el siguiente extracto de código:

//Al inicializar indigitall
indigitall.Init("<YOUR-APP-KEY>", "<YOUR-SENDER-ID>", true);

Validar la integración

Para comprobar que la integración se ha realizado correctamente realiza lo siguiente:

1. Desde Visual Studio, ve a la **pestaña de log** y busca la llamada PUT /device conteniendo los parámetros *appKey*, deviceId y pushToken y que devuelva HTTP 200.

https://youtu.be/kUqXNsN8Egc

2. Envía una notificación desde la consola de indigitall. Es posible que en la consola el contador de dispositivos aparezca a 0. No te preocupes, puede tardar unos minutos en actualizarse, pero no hace falta que esperes, la push debería llegar igualmente.

https://youtu.be/2BekYNAbwsg

3. INTEGRACIÓN PUSH WEB

Integración JavaScript Guía de integración rápida del SDK de JavaScript.

Tabla de contenidos

- ¿Qué necesitas para la integración?
- Integración
- Importando los ficheros manualmente
- Usando npm
- <u>(Opcional) Integración para Safari</u>
- Validar la integración

¿Qué necesitas para la integración?

- <u>Crear una cuenta</u> de indigitall
- <u>Crear un proyecto</u> en tu cuenta de indigitall
 - Necesitarás el **App Key** del proyecto. Lo puedes encontrar en la <u>consola de</u> <u>administración</u> dentro de la sección *Configuración* en la pestaña *Proyectos*
- (Opcional) Y para dar soporte a <u>Safari</u>:
 - Un *certificado .p12* obtenido desde la <u>consola de Apple</u>. Consulta las <u>instrucciones para obtener el .*p12*</u>
 - Hay que implementar un botón para solicitar el permiso de las notificaciones.
 Puedes encontrar más información <u>aquí</u>

Integración

Este artículo muestra el desarrollo mínimo que hay que hacer para comenzar a pedir el permiso de notificaciones y poder realizar las primeras campañas push.

Atención: no se puede pedir el permiso de notificaciones integrando el código dentro de un *iframe*

Hay dos maneras disponibles para añadir el SDK a tu proyecto. Puedes usar el gestor de dependencias <u>npm</u> o añadir los ficheros manualmente.

Integración importando los ficheros manualmente

Puedes ver nuestro vídeo tutorial o leer el texto más abajo:

https://youtu.be/o kOpQNdM0Y

1. Descarga y descomprime el .zip que contiene los dos ficheros del SDK (sdk.min.js y worker.min.js)

DESCARGAR SDK

2. Agrega ambos ficheros a tu proyecto copiándolos en la carpeta de recursos JavaScript. No te preocupes, la ruta es personalizable.

Atención: los ficheros deben alojarse en el mismo dominio y subdominio donde se vaya a integrar el servicio

3. Añade el siguiente código en todas las páginas que puedan solicitar el permiso de notificaciones. Revisa y personaliza los valores de src y workerPath.

```
<script

<rc="/indigitall/sdk.min.js"

onload="indigitall.init({

appKey: 'xxxxxxx-xxxx-xxxx-xxxx-xxxxx,

workerPath: '/indigitall/worker.min.js',

requestLocation: true

})"

async>

</script>
```

Recuerda: no olvides reemplazar el App Key por el de tu proyecto

Integración con npm

<u>npm</u> (Node Package Manager) es un sistema de gestión de paquetes. Consiste en un cliente de línea de comandos y una base de datos en línea de paquetes públicos y privados.

Puedes ver nuestro vídeo tutorial o leer el texto más abajo: https://youtu.be/BR-k4a3j4a0

1. Incluye el SDK como dependencia en tu fichero package.json.

```
...
"dependencies": {
...
"indigitall-webpush": "^3.1.3"
}
...
```

Atención: los ficheros deben alojarse en el mismo dominio y subdominio donde se vaya a integrar el servicio

2. Añade el siguiente código en todas las páginas que puedan solicitar el permiso de notificaciones.

<script< td=""></script<>
<pre>src="/node_modules/indigitall-webpush/sdk.min.js"</pre>
onload="indigitall.init({
appKey:'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
workerPath:'/node_modules/indigitall-webpush/worker.min.js',
requestLocation:true
<pre>})"</pre>
async>

Recuerda: no olvides reemplazar el App Key por el de tu proyecto

Integración para Safari (opcional)

Para poder realizar la integración para *Safari* es necesario implementar un botón para solicitar el permiso de recepción de notificaciones. A continuación, puedes ver un ejemplo de código:

```
<script>
var indigitallParams = {
  appKey: "xxxxxxxx-xxxx-xxxx-xxxx-xxxx,,,
  workerPath: '/indigitall/worker.min.js'
};
function onIndigitallLoaded(){
  indigitall.init(indigitallParams);
  if(indigitall.isSafariPushNotificationSupported()){
  var button = document.createElement("span");
   button.innerHTML = `<div id="indigitall-button"</pre>
onclick="indigitall.init(indigitallParams);document.getElementById('indigitall-
button').remove();" style="position: fixed; bottom: 15px; left: 15px; z-index: 999; transition:
.2s ease;">
    <a style="display: block; border-radius: 50%; border: 1px solid #74d227; width: 50px;
height: 50px; color: #fff; background: linear-gradient(to bottom,#89da3e 1%,#6dbf37 100%);
text-align: center; vertical-align: middle; line-height: 60px; cursor: pointer; box-shadow: 0 0
3px #000; transition: .35s ease; transition-property: all; font-size: 20px; text-shadow: 0 0 2px
rgba(0,0,0,.5); position: relative;">
```

<svg style="width:24px;height:24px" viewBox="0 0 24 24">



Validar la integración

Para probar que la integración se ha realizado correctamente, deberías ser capaz de registrar tu navegador en el servicio de notificaciones y comprobar que no hay errores. Te lo contamos en dos sencillos pasos:

1. Entra en tu web y acepta el permiso de notificaciones.



2. Busca en el log del navegador la llamada PUT /device conteniendo los parámetros appKey, deviceId y pushToken, y que devuelva HTTP 200

3. Envía una notificación desde la consola de indigitall. Es posible que en la consola el contador de dispositivos aparezca a 0. No te preocupes, puede tardar unos minutos en actualizarse, pero no hace falta que esperes, la push debería llegar igualmente.

https://youtu.be/2BekYNAbwsg

Integración Tag Manager Guía de integración rápida con TagManager. Tabla de contenidos

<u>¿Qué necesitas para la integración?</u> Integración Validar la integración

¿Qué necesitas para la integración?

<u>Crear una cuenta</u> de indigitall <u>Crear un proyecto</u> en tu cuenta de indigitall Necesitarás el app key del proyecto. Lo puedes encontrar en la <u>consola de</u> <u>administración</u> dentro de la sección Configuración en la pestaña Proyectos

Integración

Esta integración hay realizarla alojando los ficheros del SDK en tu servidor, para luego importarlos mediante tu Tag Manager.

Atención: los ficheros deben alojarse en el mismo dominio y subdominio donde se vaya a integrar el servicio

Puedes ver nuestro vídeo tutorial o leer el texto más abajo:

https://youtu.be/lyme-ZK9irQ

Pasos a seguir

- Descarga y descomprime el .zip que contiene los dos ficheros del SDK (sdk.min.js y worker.min.js) DESCARGAR SDK
- 2. Agrega ambos ficheros a tu proyecto copiándolos en la carpeta de recursos JavaScript. No te preocupes, la ruta es personalizable.
- 3. Creación de un nuevo tag desde la consola de Google Tag Manager.



4. Accede a la Configuración del Tag.

× inc	ligitall 🗅	×	Elegir un tipo de etiqueta	Q
	Configuración de la etiqueta	D	Descubra más tipos de etiquetas en la galería de plantillas de la comunidad	>
		0	Conversion Linker Google	
		÷	Google Optimize Google Marketing Platform	
		-	Google Surveys Website Satisfaction Google Marketing Platform	
	Elija un tipo de eltiqueta para empezar con la Más información	Pers	onalizado	
		$\langle \rangle$	Custom HTML Custom HTML Tag	
	Activacion	-	Custom Image Custom Image Tag	
		Más		
		AB	AB TASTY Generic Tag AB TASTY	
	Elija un activador para esta etiqu Más información	¥	Adometry Adometry	
		٥	AdRoll Smart Pixel AdRoll	
		~	Audience Center 360	

5. Añade las siguientes líneas de código en la parte de *Configuración del Tag*. Revisa y personaliza los valores de src y workerPath.

Atención: no olvides reemplazar el App Key por el de tu proyecto

```
<script>
(function() {
var el = document.createElement('script');
el.src = '/indigitall/sdk.min.js';
el.async = 'true';
el.addEventListener('load', function(){
indigitall.init({
appKey:'xxxxxx-xxxx-xxxx-xxxx-xxxx-xxxxx,
workerPath:'/indigitall/worker.min.js'
});
});
});
document.head.appendChild(el);
})();
</script>
```

En la consola de Google Tag Manager quedaría de esta forma:

Cor	figuración de la etiqueta	
Tipo	de ettqueta	
	Custom HTML Custom HTML Tag	/
нтм	L 0	
1	<pre><script> (function() {</pre></th><th></th></tr><tr><td>4</td><td><pre>el.src = '/indigitall/sdk.min.js';</pre></td><td></td></tr><tr><td>5</td><td>el.async = 'true';</td><td></td></tr><tr><td>2</td><td><pre>el.addbventListener('load', function(){ indigitall.init({</pre></td><td></td></tr><tr><td>8</td><td>appKey: '765b4222-ae48-xxxx-80e2-213c62f337df',</td><td></td></tr><tr><th></th><th>workerPath: //indigitall/worker.min.js'</th><th></th></tr><tr><th>11</th><th>));</th><th></th></tr><tr><th></th><th>document.head.appendChild(el);</th><th></th></tr><tr><th>1.3</th><th>//(); </script></pre>	
	И	
L U	Support document.write 💮	

6. En la parte de *Elegir un activador* selecciona All Pages .

\times	Elegir un activador 1 seleccionados			Q	+	Añadir
~	Nombre 1	Tipo	Filtrar			
~	All Pages	Page View				

7. Dale al botón Guardar (arriba a la derecha).

Custom HTML Custom HTML Tag	
<pre>HTML ③ i carript> 2 (functio() { 3 to intro i document.cicestellament('script'); 4 to intro i 'indigitall/adk.min.js'; 5 to intro i 'indigitall/adk.min.js'; 6 to indigitall.init({ 7 to indigitall.init({ 7 to indigitall.init({ 7 to indigitall/vorter.min.js' 8 to indigitalll/vorter.min.js' 8 to indigitall</pre>	
Activación Activadores	+
	_

Validar la integración

Previsualiza y publica el nuevo Tag. En tu página de Google Tag Manager debes tener algo parecido a esto:

Cambios del espacio de trabajo							
Nombre 个	Tipo	Cambio	Última modificación				
indigitall	Etiqueta	Añadido	hace unos segundos				

Para probar que la integración se ha realizado correctamente, deberías ser capaz de registrar tu navegador en el servicio de notificaciones y comprobar que no hay errores. Te lo contamos en dos sencillos pasos:

1. Entra en tu web y acepta el permiso de notificaciones.



Busca en el log del navegador la llamada PUT /device conteniendo los parámetros *appKey*, *deviceId* y *pushToken*, y que devuelva *HTTP 200*

[IND]Client: Method: PUT

Envía una notificación desde la consola de indigitall. Es posible que en la consola el contador de dispositivos aparezca a 0. No te preocupes, puede tardar unos minutos en actualizarse, pero no hace falta que esperes, la push debería llegar igualmente. https://youtu.be/2BekYNAbwsg

Integración WordPress

Guía de instalación rápida del plugin de WordPress.

Tabla de contenidos

- ¿Qué necesitas para la integración?
- Integración
- Validar la integración

¿Qué necesitas para la integración?

- <u>Crear una cuenta</u> de indigitall
- <u>Crear un proyecto</u> en tu cuenta de indigitall
- Acceso al panel de administración de tu WordPress
- Necesitarás permisos para instalar plugins

Integración

Gracias a la usabilidad que proporciona WordPress no necesitarás conocimientos técnicos para integrar indigitall en tu web.

Podrás encontrar nuestro plugin en el <u>Marketplace de WordPress</u>, o en el buscador de plugins que hay en panel de administración de WordPress.

1. Accede a tu panel de administración de Wordpress, navega hasta Plugin > Añadir Nuevo, busca indigitall e instala el plugin.





Añadir plugins	Subir plugin			
Resultados de la búsque	eda Destacados	Populares	Recomendados	Favoritos
Palabra clave 🗘 indig	itall			
C indigitall	indigitall WebF Activo Más d indigitall is an intern offering a SaaS Mar service that allows o enriched, segmente Por indigitall	Push etalles ational compar keting Automar pur customers s ed, int	ny that is tion sending	

2. Una vez instalado, navega a la sección Plugin > Plugins Instalados y activa el plugin de indigitall.

Plugins Añadir nuevo	
Todo (1) Activo (1) Acciones en lote 🗘 Aplicar	
Plugin	Descripción
Indigitall Web Push Notifications Desactivar	indigitall is an international company that is offering a SaaS customers sending enriched, segmented, interactive and lo audience.
	Versión 1.8.0 Por indigitall Ver detalles
Plugin	Descripción
Acciones en lote 🖨 Aplicar	

3. Aparecerá una nueva sección en su barra lateral con el menú indigitall. Accede al menú y sigue las instrucciones.

Webpush setup							
Follow these steps to add Web Push to your Wordpress blog Web Push Diverview							
1	Create an indigital account or log in your existing account. This step is essential for the proper functioning of the plugin.	Create account	Login existing account				
2	Log in your e-meil account and choose your project or create it.	Create Proyect	Select Proyect				
3	Learn how to use campaign panel and testing, Learn in a few minutes how to use the panel and create a proview for a testing, "If you need further information check our Documentation		Enable webpush				
	TROUBLESHOOTING						
(?)	If you need extra information, you can check our Wordpress Guidance						
	If you have any more doubts, don't healtate to contact us in info@indigital.com						

Introduce tu usuario y contraseña de indigitall Elige el proyecto de indigitall que usarás para gestionar tu web

Validar la integración

Para probar que la integración se ha realizado correctamente, deberías ser capaz de registrar tu navegador en el servicio de notificaciones y comprobar que no hay errores. Te lo contamos en dos sencillos pasos:

1. Entra en tu web y acepta el permiso de notificaciones.



2. Busca en el log del navegador la llamada PUT /device conteniendo los parámetros *appKey*, *deviceId* y *pushToken*, y que devuelva *HTTP 200*

3. Envía una notificación desde la consola de indigitall. Es posible que en la consola el contador de dispositivos aparezca a 0. No te preocupes, puede tardar unos minutos en actualizarse, pero no hace falta que esperes, la push debería llegar igualmente.

https://youtu.be/2BekYNAbwsg