



**Integra y automatiza el  
proceso de firma en tu negocio.**

Tecnología de firma electrónica avanzada, segura, ágil y sencilla.

# Integración SDK SIGNply iOS



versión 0.3.5

<b>1. Introducción</b>	<b>2</b>
<b>2. Contenido</b>	<b>2</b>
<b>3. Requisitos previos</b>	<b>2</b>
<b>4. Utilización</b>	<b>3</b>
4.1 Parámetros principales de llamada	3
4.2 Parámetros de SIGNply	4
4.3 Resultados de SIGNply	8

## 1. Introducción

Bienvenid@ a la documentación de la integración del **SDK SIGNply**. Por medio de este SDK podrá añadir la funcionalidad de firmar un fichero **PDF** sin apenas esfuerzo. En el presente documento se incluye el manual de integración del SDK para perfiles técnicos.

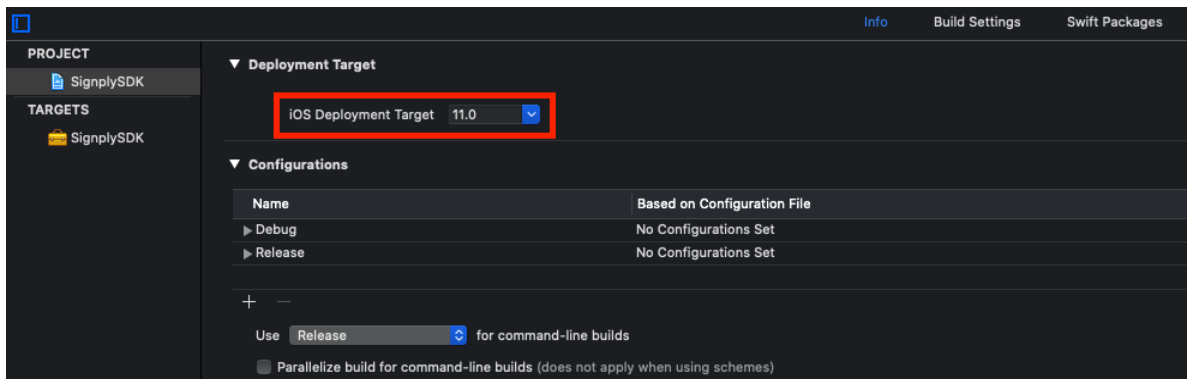
El SDK incluye los siguientes **idiomas**: *español, inglés y portugués*.

## 2. Contenido

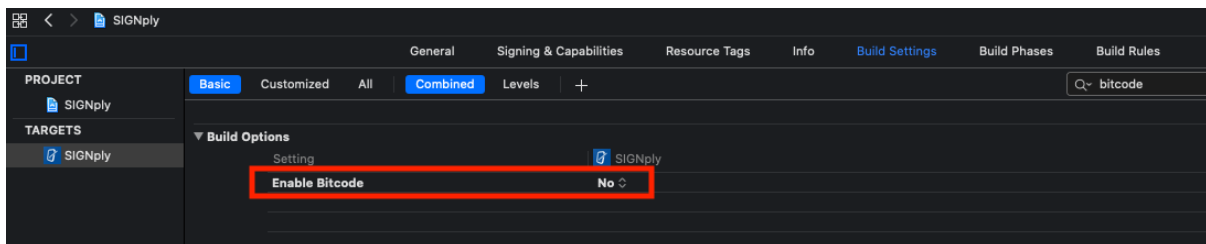
- **SDK**: Se facilita el SDK por medio de una carpeta llamada *SignplySDK.xcframework* compatible con dispositivos físicos y con los simuladores
- **Licencia**: Se facilita un archivo de licencia en base64 para poder utilizar el SDK
- **Manual de integración**: Hace referencia a este documento
- **Proyecto de ejemplo Swift**: Ejemplo completo de llamada al *SDK* de *SIGNply* para poder facilitar la integración. Puede servir cómo un punto de entrada para el proyecto a desarrollar. Contiene una licencia demo.
- **Proyecto de ejemplo Objective C**: Ejemplo simple de llamada al SDK de SIGNply desde un proyecto en Objective C. Contiene una licencia demo.

## 3. Requisitos previos

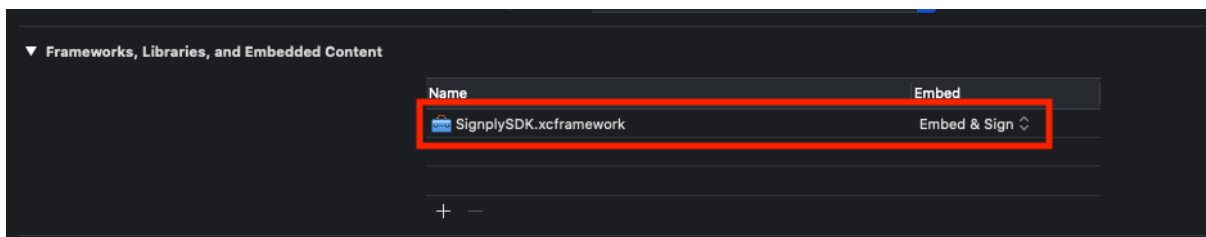
Este *SDK* es compatible desde la versión de *iOS 11* tanto con simuladores como con dispositivos físicos (iPhone y iPad). Es necesario establecer la *versión mínima 11.0* como se muestra en la imagen inferior.



Para una correcta compilación, el *bitcode* del proyecto debe estar *desactivado*. Se muestra en la imagen inferior.



Añadir el framework es muy sencillo, tan solo es necesario arrastrar la carpeta *SignplySDK.xcframework* al apartado de Frameworks, Libraries and Embedded content. El resultado se muestra en la imagen inferior.



## 4. Utilización

En el código es necesario importar el SDK por medio de `import SignplySDK` al principio de cada archivo donde se vaya a utilizar.

El SDK de SIGNply es lanzado a través del método `SignplySDKLauncher.handleDocument`

Este método recibe varios parámetros configurables y devuelve un `successCallback` en el caso de que el documento haya sido firmado correctamente, un `rejectCallback` en el caso de que el usuario rechace el documento y un `errorCallback` en el caso de que se haya producido algún error en el proceso de firmado.

### 4.1 Parámetros principales de llamada

SIGNply dispone de una serie de parámetros de llamada para la configuración del firmado de documento que se exponen a continuación:

**signplySDKParams** (`SignplySDKParams`): estructura de datos que incluye toda la configuración necesaria.

**on** (`UIViewController`): controlador utilizado para mantener la navegación correcta y poder instanciar el controlador de firma.

**successCallback** (`URL`, `String`) -> (`()`): Devolución de llamada con la url del documento o el documento en base64 (depende de si en la llamada se le pasa la url del documento o el documento en base64).

**rejectCallback** (`String`) -> (`()`): Devolución de llamada con el motivo de rechazo del documento en el caso de que se haya producido.

**errorCallback** (`String`) -> (`()`): Devolución de llamada con el mensaje de error en el caso de que se haya producido.

### 4.2 Parámetros de SIGNply

Para facilitar la integración, la mayoría de parámetros son opcionales. El objeto `SignplySDKParams` se puede formar tan solo con esta mínima configuración en Swift.

```
SignplySDKParams(licenseB64: licenseB64, documentB64: documentB64,  
widget: SignplySDKWidget(widgetType: .manual))
```

A continuación se muestran todos los parámetros disponibles y configurables, cabe destacar que los `init` de cada clase pueden cambiar entre los tipos de objeto `Int` ->

*NSNumber* o también *Bool* -> *NSNumber*. Esto se ha realizado así para mantener la compatibilidad con Objective C.

### **SignplySDKParams :**

- **licenseB64** (String) : licencia de ESS en base64
- **documentB64** (String): documento en base64
- **documentPath** (URL): ruta absoluta del documento. Prevalece sobre el documentB64.
- **saveDocumentSignedName** (String): nombre de documento firmado.
- **widget** (SignplySDKWidget): widget de firma configurable
- **tsp** (SignplySDKTSP): TSP configurable
- **ocsp** (SignplySDKOCSP): OCSP configurable
- **header** (SignplySDKHeader): parámetros de la cabecera de la firma
- **extra** (SignplySDKExtra): funcionalidad extra en el SDK configurable
- **certificate** (SignplySDKCertificate): certificado de firma configurable

### **SignplySDKWidget**

- **widgetType** (SignplySDKWidgetType): tipo de posicionamiento del widget
- **widgetFloatText** (String): indica la cadena a buscar dentro del documento donde se incrusta el widget. Devuelve el primer resultado obtenido. Este parámetro es sensible a mayúsculas y a minúsculas.
- **widgetFieldFieldname** (String): nombre de campo preexistente en el que se incrusta el widget.
- **widgetManualRatio** (Float): relación entre el ancho y el alto de widget para el posicionamiento manual. Por defecto 2,5.
- **widgetFixedPage** (NSNumber): número de página en el que se incrusta el widget.
- **widgetFixedX** (NSNumber): indica el desplazamiento en horizontal de la posición del widget desde el vértice inferior izquierdo de la página de un documento.

- **widgetFixedY** (NSNumber): indica el desplazamiento en vertical de la posición del widget desde el vértice inferior izquierdo de la página de un documento.
- **widgetFloatGapY** (NSNumber): indica el número entero de unidades de desfase (positivo o negativo) en vertical desde la parte inferior al texto
- **widgetFloatGapX** (NSNumber): indica el número entero de unidades de desfase (positivo o negativo) en horizontal desde la parte inferior al texto
- **widgetCustomText** [SignplySDKWidgetCustomText]: array de texto multilínea que se incrusta en el widget de firma
- **widgetRequestCustomText** (Bool): indica si se le debe pedir al usuario una cadena de texto para incrustar en el widget de firma en el momento de realizarla.
- **widgetWidth** (Int): ancho del widget
- **widgetHeight** (Int): alto del widget
- **flexible** (Bool): Variable que hace más flexible las validaciones de los parámetros si es true. Recomendación tenerla a false.
- **signatureColor** (UIColor): Color de la firma
- **signatureThickness** (Int): Grosor de la firma. Recibe valores entre 1 y 5.

### SignplySDKTSP

- **tspActivate** (Bool): activación de introducción de sello de tiempo
- **tspURL** (String): url de TSP
- **tspUser** (String): usuario de TSP
- **tspPasswordB64** (String): password de TSP en base64

### SignplySDKOCSP

- **ocspActivate** (Bool): activación de introducción de sello de tiempo

- **ocspURL** (String): url de OCSP
- **ocspUser** (String): usuario de OCSP
- **ocspPasswordB64** (String): password de OCSP en base64

### **SignplySDKHeader**

- **author** (String): Autor de la firma
- **reason** (String): Razón de la firma
- **contact** (String): Contacto del autor o entidad de la firma
- **location** (String): Localización del autor o entidad de la firma

### **SignplySDKExtra**

- **autoOpen** (Bool): Permite desplegar el widget de firma automáticamente al abrir el documento
- **captureSignatureSeconds** (Int): Tiempo previo a que se cierre el widget de firma
- **showCaptureSignatureSeconds** (Bool): Muestra el tiempo previo a que se cierre el widget de firma
- **viewLastPage** (Bool): Si es true, es necesario que el usuario visualice hasta la última página para poder firmar
- **certify** (Bool): Certifica el documento
- **documentPassword** (String): Desbloquea el documento en el caso de que esté protegido por contraseña
- **showReject** (Bool): Muestra el botón de rechazar documento

### **SignplySDKCertificate**

- **signCertP12B64** (String): Certificado de firma en base64. Si se envía nula toma el valor por defecto.



- **signCertPasswordB64** (String): Contraseña del certificado de firma en base64. Si se envía nula toma el valor por defecto.
- **encKeyB64** (String): Clave pública de cifrado de los datos biométricos en base64. Si se envía nula toma el valor por defecto.

### **SignplySDKWidgetCustomText**

- **fontSize** (Int): Tamaño de fuente del texto personalizado
- **text** (String): Valor del texto personalizado

### **SignplySDKWidgetType**

- **manual**: Posicionamiento modo manual en el cual el usuario puede mover la firma libremente (*en Objective C toma el valor 0*).
- **field**: Posicionamiento que busca un campo de firma en concreto (*en Objective C toma el valor 1*).
- **fixed**: Posicionamiento fijo en el documento, se le pasa un número de página y una posición relativa dentro del documento (*en Objective C toma el valor 2*).
- **float**: Posicionamiento flotante en el documento en base a una búsqueda de cadena de texto (*en Objective C toma el valor 3*).

## 4.3 Resultados de SIGNply

Signply devuelve resultados por medio de callbacks para simplificar la integración. En la llamada al método `SignplySDKLauncher.handleDocument` se le pasan tres callbacks:

**successCallback**: Devuelve la *URL* del documento firmado y el documento en *base64*. La url del documento es temporal con lo que se recomienda copiar el archivo a otra ubicación en el caso de necesitar trabajar en un futuro con él.

**rejectCallback**: Devuelve un String con la razón del rechazo del documento.

**errorCallback**: Devuelve un *String* con el error producido en el proceso de firmado en el caso de que se haya producido alguno.