



**DESARROLLO DE UN
CLIENTE SOFIA2 (KP)
SIGUIENDO
ARQUITECTURA
KP-MODELO**

JUNIO 2014

Versión 1

INDICE

INDICE.....	2
PASO 1 - FAMILIARIZARSE CON LOS CONCEPTOS BÁSICOS DE SOFIA2	3
PASO 2 - OBTENER CREDENCIALES DE UTILIZACIÓN DE SOFIA2.....	4
REGISTRO EN LA PLATAFORMA SOFIA2	4
LOGIN EN SOFIA2 ON CLOUD	5
PASO 3 - SELECCIONAR PLATAFORMA DE DESARROLLO (WINDOS, MAC, LINUX) Y DESCARGARSE EL SDK.	7
DESCARGA E INSTALACIÓN DEL SDK DE SOFIA2.	7
ARRANQUE DEL SDK DE SOFIA2.....	8
IDE JAVA.....	9
CONSOLA DE ARQUITECTURA.....	10
PASO 4- CONFIGURACIÓN.....	12
ALTA DE KP EN LA PLATAFORMA.....	12
GENERACIÓN DE TOKEN.....	12
CREAR UNA NUEVA CONFIGURACIÓN.....	15
ASIGNACIÓN DE SOFTWARE EN KPS.....	17
PASO 5- DESARROLLO DE UN KP PARA UN GATEWAY SIGUIENDO EL KP MODELO.	18



Paso 1 - Familiarizarse con los conceptos básicos de SOFIA2

- **Smart Space:** Entorno colaborativo virtual donde dispositivos y aplicaciones interoperan para ofrecer una funcionalidad compleja.
- **KP (Knowledge Processor):** Representa cada cliente en un Smart Space produciendo y/o consumiendo información.
- **SIB (Semantic Information Broker):** Es el core del Smart Space, actúa como elemento integrador de la información semántica intercambiada y como almacén de información
- **SSAP (Smart Space Access Protocol):** Es el protocolo estándar de mensajería entre los KPs y el SIB.
- **Ontología:** Representa una definición de Entidad dentro de mi Sistema, existen predefiniciones (para ámbito Ciudad por ejemplo), también pueden crearse.
- **Instancia de Ontología:** Representa una Entidad concreta en el Sistema, es una estructura JSON.
- **KPModelo:** Modelo de desarrollo de KP gestionados por el SIB, basado en Workers en un modelo de publicación suscripción.

Paso 2 - Obtener credenciales de utilización de SOFIA2

El proceso para desarrollar sobre la plataforma SOFIA es el siguiente:



Registro en la plataforma SOFIA2

Accediendo a la página <http://sofia2.com/console/gestionusuarios/registerUser> podemos darnos de alta en la plataforma ROL_USUARIO, lo que nos permitirá suscribirnos a ontologías públicas y realizar consultas sobre ella.

Solicitud Alta de Usuario

Formulario

Usuario

Password

Nombre

Email



Será necesario enviar un correo al Administrador de la plataforma para solicitar el cambio a ROL_COLABORADOR, ya que es necesario realizar una serie de operaciones a las cuales no tendría acceso.

Login en SOFIA2 On Cloud

En la página <http://sofia2.com/console/login> nos logaremos con el usuario creado en el punto anterior, lo que nos proporcionará acceso a la consola

USUARIOS

PLANTILLAS

ONTOLOGÍAS

Crear Ontología

Crear Ontología con Wizard

Listar Ontologías

Autorizaciones Ontologías

Crear Ontología Grupo

Listar Ontologías Grupo

Autorizaciones Ontologías Grupo

Crear Ontología desde Excel

ASSETS

KP's

CEP

CONSULTAR BDTR

SCRIPTS

HERRAMIENTAS

DESPLIEGUE

API MANAGER

Bienvenido a Consola Web SOFIA2

Pasos para empezar a trabajar con la plataforma

✓ Suscripción a Ontologías: Paso 1

✓ Creación de Kp de la ontología: Paso 2

✗ Tokens del Kp: [Crear Token](#) Paso 3

[Descargar ejemplo Kp Java](#)

[Descargar ejemplo Kp Javascript](#)

122.075.589

Mensajes procesados por Sofia2InCloud

INDRA 2014 Versión 2.9



Paso 3 - Seleccionar plataforma de desarrollo (Windows, Mac, Linux) y descargarse el SDK.

Descarga e Instalación del SDK de SOFIA2.

Descargar la última versión del Entorno desde las siguientes URLs:

- SOFIA2 SDK (Windows)

http://sofia2.org/sdk/SOFIA2_SDK_2.6_WIN.zip

- SOFIA2 SDK (MAC)

http://sofia2.org/sdk/SOFIA2_SDK_2.6_MAC.zip

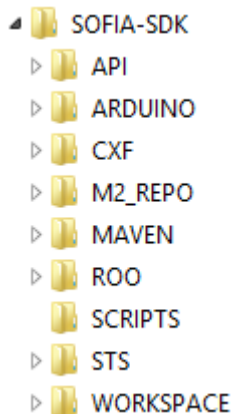
- SOFIA2 SDK (LINUX)




http://sofia2.com/sdk/SOFIA2_SDK_2.6_LINUX.tar.gz

Descomprimir el ZIP descargado:



Dentro del directorio **SOFIA-SDK** encontramos todos los componentes que componen el SDK.



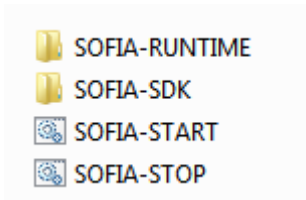
- Los **entornos de desarrollo** se encuentran en los directorios  ARDUINO y  STS para los entornos C de Arduino y Java respectivamente.
- En el directorio  ROO encontramos la consola de Arquitectura que dispone los comando sofia para la creación de Plugins, Kp's y Gateway's

- En el directorio CXF encontramos las herramientas de productividad para trabajar con WSDL y WADL.
- En los directorios MAVEN y M2_REPO se encuentran la Herramienta Maven las dependencias manejadas por esta para el desarrollo sobre la Plataforma Sofia.
- En el directorio API encontramos las API Java y Javascript para el desarrollo de KP.
- En el directorio WORKSPACE encontramos los directorios de configuración de los IDE.
- En el directorio SCRIPTS se disponibilizan los procesos por lotes de los diferentes componente que forman el SDK.

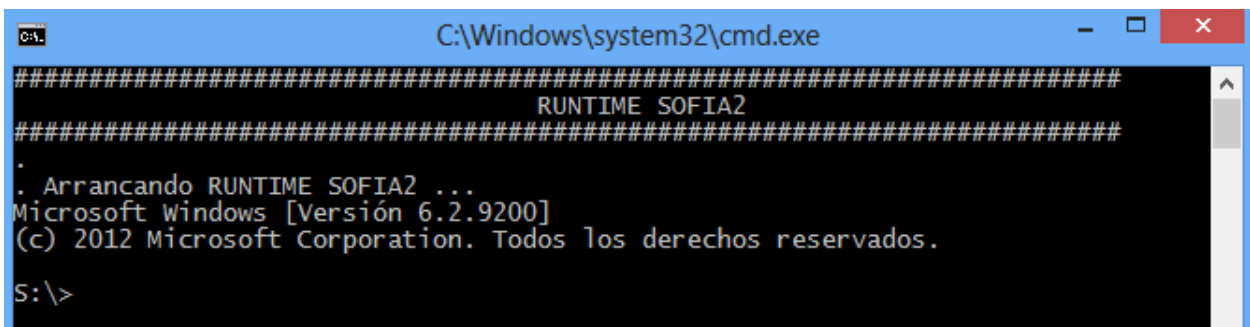
- ARDUINO
- ARQSPRING
- ARQSPRINGWSDL2JAVA
- IDE

Arranque del SDK de SOFIA2


Una vez descargado el SDK de SOFIA2 en el directorio raiz de SOFIA encontramos el siguiente contenido.




Haciendo doble clic sobre el proceso por lotes SOFIA-START se creará la unidad Virtual **S:**, y se abrirá la consola de comandos.



La sesión de esta consola de comandos está configurada para operar con el Runtime y SDK de Sofia.

La instalación del SDK de SOFIA2, modifica el fichero  **SOFIA-START** creado por el Runtime, por lo que es importante que se respeten los directorios de instalación.

IMPORTANTE

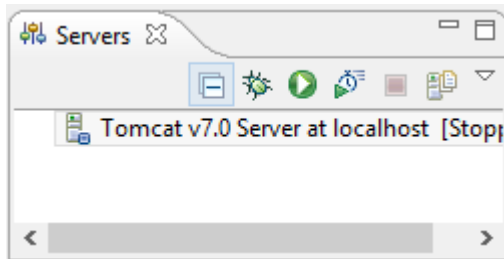
Si se cierra la consola de comandos deberemos volver a ejecutar el proceso por lotes  **SOFIA-START** para que se vuelva a abrir la consola con la configuración predeterminada.

IDE Java

Ejecutando en la Consola el comando **S:\>ide**

```
S:\>ide
#####
                        SDK SOFIA2
#####
. Arrancando IDE...
.
```

Se abrirá el **IDE Eclipse de la Arquitectura iTR de Indra** para el desarrollo de aplicaciones SOFIA2.



Consola de Arquitectura

Ofrece herramientas de productividad para crear proyectos SOFIA2 de forma sencilla.

Ejecutando en la Consola el comando **S:\>arqspring**

```
S:\>arqspring
#####
                        SDK SOFIA2
#####
. Arrancando ArqSpring...
```

Se abrirá la **consola de Arquitectura iTR**



Paso 4- Configuración.

Alta de KP en la plataforma

Un usuario deberá registrar en la plataforma sus KPs, de lo contrario, la plataforma rechazará la conexión de los mismos.

Para registrar un KP, la plataforma proporciona la sección **Gestion KPs**, donde un usuario podrá crear un nuevo KP o administrar los que ya tiene dados de alta:

USUARIOS

PLANTILLAS

ONTOLOGÍAS

ASSETS

KP's

Crear KP

Listar KPs

Gestión Tokens

KP's Activos

Gestión de Instancias de KP's

CEP

CONSULTAR BDTR

SCRIPTS

HERRAMIENTAS

DESPLIEGUE

API MANAGER

KPs > Crear KP

Crear Nuevo KP

Formulario

Identificación Clave de cifrado
48412f7e-5cc6-4b9a-977e-4bf5a7fd7292

Descripción

Ontología Ontologías de Grupo

Alarma
Alarma2
alertCuadroElectrico
Anuncios

Borrar
Imgracia_pru_1
Prueba
Basuras

Meta-Información

Cancelar Crear

Como vemos, un KP podrá hacer uso de una o varias ontologías, siendo esta la información que producirá o consumirá de la plataforma.

Una vez registrado en la plataforma, el KP ya podrá establecer conexiones con la misma.

Generación de Token.

Una vez registrado en la plataforma, se necesitará generar un token para realizar conexiones con ese KP.

Para ello iremos a la sección **KPs > Gestión de Tokens**, donde el usuario podrá crear uno o varios tokens asociados a un KP.

KPs > Gestión Tokens

Gestión de Tokens

Formulario

Identificación del KP <input type="text"/>	Propietario del KP <input type="text"/>
<input type="button" value="Buscar"/> <input type="button" value="Cancelar"/>	

Listado de KPs

Search:

Identificación del KP	Propietario del KP
KPpubliAnuncios	sofia

Showing 1 to 1 of 1 entries

Listado de Tokens

No data available in table	
----------------------------	--

Para generar nuevos Tokens seleccionaremos el KP para el que queremos generarlos y el número de Tokens que queremos generar y pulsaremos el botón:

Ahora aparecerá una nueva tabla con los Tokens que hemos generado, y que podrán utilizar los KP para conectarse a la plataforma.

Identificación del KP	Propietario del KP
KPTestMeteo	usuario

Showing 1 to 1 of 1 entries

First Previous **1** Next Last

Listado de Tokens

Nombre del KP	Propietario	Token	Ult. Conexion	Activo	Eliminar
KPTestMeteo	eb193ec4-969d-43c9-9a7e-8f82cbf0b0b6	7f477b870f5c4016937aa3eb236ce66c		<input checked="" type="checkbox"/>	
KPTestMeteo	eb193ec4-969d-43c9-9a7e-8f82cbf0b0b6	f7e5bed1ae5742acb895bf2d01f55237		<input type="checkbox"/>	
KPTestMeteo	eb193ec4-969d-43c9-9a7e-8f82cbf0b0b6	bc2e66b0676242cd99af10ee398c61f3		<input type="checkbox"/>	

3

First Previous Next Last

Por defecto los Tokens están desactivados, pulsando el check Activo, los activaremos.

Confirmación

Token activado

OK

Search:

Nombre del KP	Propietario	Token	Ult. Conexion	Activo	Eliminar
KPTestMeteo	eb193ec4-969d-43c9-9a7e-8f82cbf0b0b6	7f477b870f5c4016937aa3eb236ce66c		<input checked="" type="checkbox"/>	
KPTestMeteo	eb193ec4-969d-43c9-9a7e-8f82cbf0b0b6	f7e5bed1ae5742acb895bf2d01f55237		<input type="checkbox"/>	
KPTestMeteo	eb193ec4-969d-43c9-9a7e-8f82cbf0b0b6	bc2e66b0676242cd99af10ee398c61f3		<input type="checkbox"/>	

3

First Previous Next Last

Crear una nueva configuración.

El modelo basado en Kp Modelo delega la configuración de los Kp en el SIB, para ello el usuario podrá crear una nueva configuración que podrá asociar a su Kp Modelo.

El usuario creara una nueva configuración a través de la opción Despliegue -> Crear Nueva Configuración SW.

Esta pantalla se compone de dos partes principales, “**Datos Gestión de SW**” y “**Datos de la Configuración**”

En la primera parte indicaremos los datos de gestión de software como:

- **Nombre de la aplicación:** El sistema nos avisará si intentamos crear una aplicación de software con un nombre que ya existe.
- **Activa:** Nos indica si se desea activar la aplicación de software o no.
- **Aplicación:** Permite seleccionar el war con el software a utilizar por los clientes.
- **Descripción:** Descripción a incluir sobre la aplicación de software.

En la parte de datos de configuración, indicaremos las propiedades de configuración de software (por ejemplo: kp, instancia de kp, token de conexión, ip, puerto, etc.)



Crear Nueva Configuración SW

Formulario

Datos Gestión SW

Nombre Aplicación
appTemp

Versión 1 Activa

Aplicacion
Seleccionar archivo No se ha seleccionado ningún archivo

Descripción

Datos de la Configuración

Descripción

Propiedades

Propiedad	Valor	Opciones
port	8082	
ip	127.12.17.3	

Datos de la Propiedad

Propiedad

Descripcion

Valor

Nueva Propiedad

Cancelar
Guardar

Una vez creada la aplicación de software, nos mostrará el detalle con todos los datos.

Asignación de Software en KPs.

El usuario deberá establecer una correspondencia entre la nueva configuración aplicación creada y el KP e Instancia de KP creados anteriormente.

Para ello se debe dirigir a la opción de menú Despliegue -> Asignación SW a KPs y aparecerá una pantalla como la siguiente en la que deberá de introducir los valores y pulsar el botón Asignar Versión:

DESPLIEGUE > Asignación SW a KPs

Asignación Software a KPs

Formulario

Usuario

Aplicación - Versión Versión Conf.

Kp Instancia KP

Listado de Asignaciones

Search:

Kp ▲	Instancia de Kp ◆	Aplicación - Versión ◆	Versión Conf. ◆	Usuario ◆	Ops. ◆
StatKP *		SIMPLESTAT- V1	Versión -2	sofia	

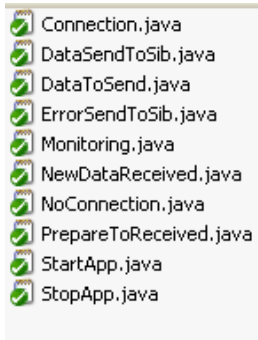
Showing 1 to 1 of 1 entries

La nueva asignación creara aparecerá en el listado de Asignaciones.



Paso 5- Desarrollo de un KP para un Gateway siguiendo el KP Modelo.

- 1) Desde la consola de la plataforma lanzamos el comando ***sofia2 crearAppModelo --id [NOMBRE APPMODELO] --paquetebase [PAQUETE APLICACION]***
- 2) Implementamos el método `readDataSensor` de la clase `PrepareToReceived`, donde deberemos de establecer la conectividad con los sensores y convertir la información obtenida de estos a un objeto `SensorMessage`.
- 3) En la clase `PrepareToReceived` podemos leer de todos los sensores o crear nuevas clases, extendiendo de `PrepareToReceivedWorkerImpl` para gestionar cada uno de los sensores conectados.
- 4) Implementaremos el método `generateSSAPMessage` de la clase `NewDataReceived`, donde debemos convertir los datos que hemos recogido de los sensores en un mensaje SSAP.
- 5) A través de la implementación de la clase `DataSendToSib` obtenemos el mensaje de respuesta del SIB junto con el mensaje original que enviamos al SIB.
- 6) Si queremos subscribirnos al sib debemos implementar la clase `SubscriptionListener` en el método `init`, podremos a través del método `subscribe(ontology, query, SSAPQueryType)`, subscribirnos al SIB, y a través del método `onEvent`, definir las operaciones a realizar cada vez que nos llega una notificación del SIB asociada a la subscripción realizada.
- 7) Adicionalmente podemos implementar el resto de clases que nos permitan controlar aspectos como el:
 - a. Arranque y Detención del servidor de la Aplicación KP. `StartApp` y `StopApp` (Este último deberá implementar el comportamiento que desamos ante una detención solicitada por el contenedor del KP).
 - b. Conexión con el SIB y pérdida de esta conectividad.
 - c. Preprocesamiento y postprocesamiento de los mensajes enviados al SIB.
 - d. Error al enviar mensajes al SIB



Los logramos implementando cada uno de los Workers definidos en el diagrama anterior.

- 8) Es posible que los desarrolladores necesiten en ocasiones Registrar nuevos workers: Estos deben ser registrados a través del bean KPWorkerCfg.

2.1 Obtener una referencia a este objeto:

```
@Autowired  
protected KPWorkerCfg kPWorkerCfg;
```

2.2 Invocar el método `public void addEvent(String event, Class message, Subscriber worker)`:

```
kPWorkerCfg.addEvent("MI_EVENTO", LifeCicleMessage.class, new StartAppWorker())
```

- 9) Para publicar un evento existente o creado por lo desarrolladores debemos, hacerlo a través del método.

`publish (String topic, Object message)` que encontramos en el bean KPWorkerCfg.

Los eventos disponibles de facto por la plataforma son los que encontramos en las clases enumeradas.

SibEvents

SensorEvents

LifeCicleEvents

InfraestructureEvents