

T 2 Video

SDK

1.0

Índice

Introducción	2
Primeros pasos	3
Diagrama	4
Nuestra solución	5
Hola Mundo	6
Acciones	8
Entrada - Salida	8
Acciones, Expresiones y Gestos	9
GetActions	10
DoActions	10
GetExpressions	12
DoExpression	12
GetGestures	14
DoGesture	14
Ask Watson	16
Speak	16
Showimage	17
Hideimage	17
Biblioteca de texturas	17
Escenario T2Video	19
Ventana de configuración	22
Ventana de creación y edición de Workspaces	25

Introducción

El **SDK** de **T2Video** es un conjunto de **herramientas de desarrollo** el cual te va a permitir desarrollar y ejecutar instancias de T2Video. Las instancias desarrolladas tienen soporte para Windows y Mac. En un futuro estaremos lanzando soporte WEB, Android y IOS.

Ofrece respaldo para la construcción de tus procesos de negocio con mecanismos de desarrollo de Software, que aportarán un plus a la interacción de los usuarios con tus Sistemas.

T2Video SDK **ofrece**:

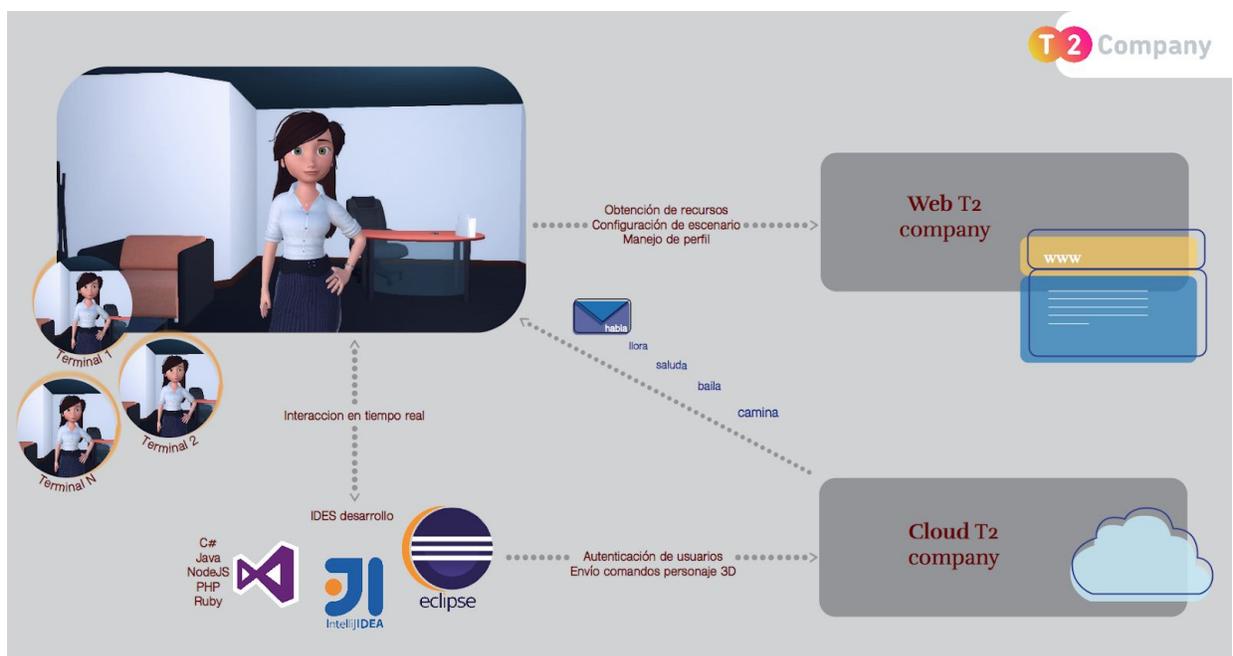
- Funciones para entablar comunicación con tus clientes
- Configuración del personaje y escenario que mostrarás
- Gestos, Acciones, Animaciones de los personajes (triste, cantar, bailar, reír)
- Tus Clientes podrán interactuar con el personaje en forma hablada o escrita

Primeros pasos

1º Ingresar a la web www.t2company.com.uy desde Windows o Mac

2º Registrarse con sus datos

3º Descargar el ejecutable (luego se podrá descargar escenarios, objetos y vestimentas para el personaje Clara)



Nuestra solución

“T2Video Platform” es un ambiente de desarrollo de intercambio con personajes animados 3D para automatizar sus procesos, brinda interacción por voz, texto y comandos, además de una capa de desarrollo para la implementación de sus flujos de negocio.

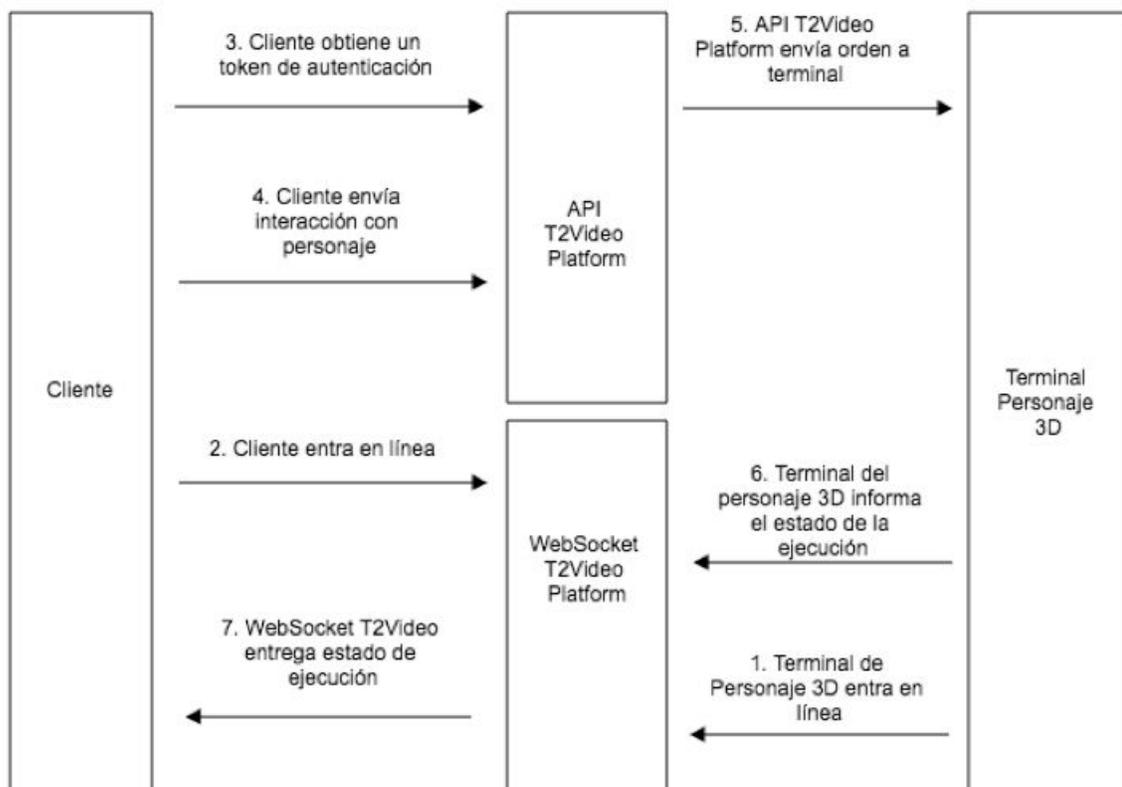


Diagrama de Secuencia T2Video Platform

Hola Mundo

1- Indicar en Terminal SDK Client Secret de desarrollo válido, para proveer de comunicación a la Terminal.

2- Conexión a WebSocket para obtener el estado de ejecución de acciones de Personaje en Terminal.

Nota: Los ejemplos que se ven a continuación, son solo a modo de guía, puede seleccionar la librería de su comodidad (en su lenguaje preferido), que ofrezca comunicación con WebSocket.

Ejemplo Lenguaje JavaScript:

```
$(function(){
  var connection = new WebSocket("wss://t2sdk.t2voice.com/ws/", ['soap', 'xmpp']);

  connection.onopen = function () {

    var jsonPresentToTerminal = {
      isClient: true,
      token: "{token_developer}",
      "characterId": "ClaraWatsonShoppingTerminal10"
    };

    connection.send(JSON.stringify(jsonPresentToTerminal));
  };

  // Log errors
  connection.onerror = function (error) {
    console.log("WebSocket Error " + error);
  };

  // Log messages from the server
  connection.onmessage = function (e) {
    console.log("Message from T2Company Terminal: " + e.data);
  };
});
```

3- Enviar petición a Servicio Rest para que el Personaje ejecute acción.

Ejemplo de llamada a Speak JavaScript, JQuery, AJAX:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

```
<script type="text/javascript">
```

```
$(function(){
    $("#claraSpeak").click(function () {

        $.ajax({
            type: "POST",
            url: "https://t2sdk.t2voice.com/actions/speak",
            data: {
                "clientSecret":"05d9dec4cb21c1a4773eab6d4737b63fd41e08e4edba0073e59217703948ba87",
                "characterId":"Clara",
                "text": "Son las 10am, ve al local más próximo que se acaba el descuento en jeans .guiniolzq",
                "params": ""
            },
            dataType: "json",
            success: function(result){
                console.log(result);
            },
            error: function(error) {
                console.log(error);
            }
        }
    )

```

```
});
```

```
});
```

```
</script>
```

```
<input type="button" id="claraSpeak" value="Clara Speak">
```

Acciones

Entrada - Salida

Todas las acciones enviadas al personaje tienen una entrada común para identificar la instancia a la que se envía la ejecución así como su comportamiento.

Parámetros comunes:

Integer id: Identifica el tipo de personaje (Género e Idioma).

String character: Identificador de la instancia a la que se envía.

Nombre	Género	Lenguaje	Identificador
Clara	Femenino	Español	1
Ema	Femenino	Inglés	2
Xuxa	Femenino	Portugués	3

Lista de characters (género, lenguaje, id)

* En caso de no enviar el parámetro id se asume Clara por defecto

Salida

Estado de la ejecución, errores en caso de ser lanzados e identificador de la acción.

Estructura:

“status”: “x message” (x = 0, 1, 2), 0 : OK, 1 : Error controlado, 2 : Error imprevisto.

“errors”: “mensaje descriptivo del error”.

“action-id”: “identificador de la acción”.

Ejemplo

Ejecución OK:

```
{“status”:0, “action-id”:“3F2504E0-4F89-11D3-9A0C-0305E82C3301”, “errors”:“”}
```

Ejecución errónea controlada

```
{“status”:1, “action-id”:“”, “errors”:“Debe autenticarse antes de hacer envíos a Clara”}
```

Ejecución errónea no controlada

```
{“status”:2, “action-id”:“”, “errors”:“500: Error interno del servicio”}
```

Acciones, Expresiones y Gestos

Acción: Animación asociada a un sonido, Ejemplo: Risas (.jaja), Aplausos (.aplaude).

Expresión: Animación de sentimiento, Ejemplo: Feliz (.feliz), Triste (.triste).

Gestos: Animaciones sin sonido, Ejemplo: Estress (.cuello, el personaje mueve cabeza y hombros relajándose).

* Se puede enviar en una secuencia la ejecución de:

Una Acción o Gesto junto expresión (Aplausos + Feliz).

Hablar junto a una expresión (“Hoy no es un buen día” + Enojada).

Hablar junto a un Gesto (“Hola cómo estás” + Saluda).

getActions

Descripción: Retorna lista de objetos (identificador, descripción) con los identificadores de las acciones soportadas por el personaje y la descripción de cada una.

Método de envío: **GET**.

Lista de acciones para el character Clara:

agradece	aplaude	ay	beso	besos	bostezo
bu	buuu	canta	chasquidolzq	chasquidoDer	ejem
emmm	emmm1	emmm2	emmm3	estornudo	grito
hey	jaja	jeje	jiji	llanto	mmm
ooh	oooh	ops	respiro	ronca	shhh
silba	suspiro	tos	trago	uff	ups
yuhooo					

doActions

Descripción: Envía una orden al personaje a ejecutar una acción.

* La ejecución de la acción interrumpe los audios generados al hablar en el transcurso de la acción.

Método de envío: **POST**.

Parámetros:

String id: Identificador de la acción a llamar a ejecutar.

Ejemplo de entrada:

```
{
  "clientSecret": "token_developer",
  "characterId": "Clara1",
  "id": "besos"
}
```



g etExpressions

Descripción: Retorna lista de objetos (identificador, descripción) con los identificadores de las expresiones soportadas por el personaje y la descripción de cada una.

Método de envío: **GET**.

Lista de expresiones:

asqueado	asustado	enojado
feliz	neutral	sorprendido
triste		

d oExpressions

Descripción: Genera una expresión facial en el personaje (por defecto feliz), la misma se acopla a los diálogos, gestos y animaciones sin modificarlos o interrumpirlos.

Método de envío: **POST**.

Parámetros:

String id: Identificador de la expresión a ejecutar.

Integer degree: Grado de intensidad que tendrá la expresión (valor entre 1...10).

Ejemplo de entrada:

```
{
  "clientSecret": "token_developer",
  "characterId": "Clara1",
  "Id": "asustado",
  "degree": "7"
}
```



getGestures

Descripción: Retorna lista de objetos (identificador, descripción) con los identificadores de los gestos soportados por el personaje y la descripción de cada uno.

Método de envío: **GET**.

Lista de gestos:

asiente	besos	bienvenida1	bienvenida2	cuello	entiende
estres	frotamanos	gesto1A	gesto1D	gesto1I	gesto1HipA
gesto1HipD	gesto1HipI	gesto2InA	gesto2InD	gesto2InI	gesto2OutA
gesto2OutD	gesto2OutI	gestoChestOutA	gestoChestOutD	gestoChestOutI	grito
guinio	guiniol	hey	manospecho	miraunias	niega
noentiende	rasca	saluda	uff	veni	yuhooo

doGestures

Descripción: Envía una orden al personaje a ejecutar un gesto.

* La ejecución de la acción no interrumpe los audios de habla en ejecución.

Método de envío: **POST**.

Parámetros:

String id: Identificador del gesto a ejecutar.

Ejemplo de entrada:

```
{
  "clientSecret": "token_developer",
  "characterId": "Clara1",
  "id": "guiniol"
}
```



askWatson

Descripción: Envía un texto como pregunta a los distintos workspaces pre cargados de Watson, el personaje expresa la contestación correspondiente a la pregunta.

Método de envío: **POST**.

Parámetros:

String question: Pregunta que se enviará a los Workspaces cargados.

Ejemplo de entrada:

```
{
  "clientSecret": "token_developer",
  "characterId": "Clara1",
  "question": "¿Cómo puedo contactarme con la empresa en caso de problemas?"
}
```

Speak

Descripción: Envía un texto al personaje para hablar, se le pueden agregar otras acciones.

Método de envío: **POST**.

Parámetros:

String text: Texto enviado al personaje para hablar.

String parameters: Parámetros de reemplazo en el texto.

float speed: Controla la velocidad con la que el personaje habla el texto (Si no se incluye se utiliza la velocidad por defecto).

Ejemplo de entrada:

```
{
  "text": "Perfecto Cliente, usted ha pago satisfactoriamente su cuenta de Servicio",
  "parameters": "Cliente:Vicente-Servicio:UTE"
}
```

También es posible utilizar las acciones, las expresiones y los gestos en el texto del Speak.

Notación actions: .[nombre de acción]. Ejemplo: .jaja

Notación gestures: :[nombre de acción]. Ejemplo: :estres

Notación expressions: :[nombre de expresión]_[grado de expresión]. Ejemplo: triste_7

ShowImage

Descripción: Muestra una imagen deseada en la escena.

Método de envío: **POST**.

Parámetros:

String url: dirección web donde se ubica la imagen a cargar.

String shape: forma de la imagen a mostrar. Estas pueden ser: tall (tamaño 1x2), wide (tamaño 2x1) y square (tamaño 1x1).

String side: posición de la imagen a mostrar. Estas pueden ser: left (izquierda), right (derecha).

Boolean cache: indica si se utiliza el caché, si el valor es 'true' la imagen será guardada en disco, si es 'false' no. (Si este campo es omitido la imagen queda guardada).

hidelimage

Descripción: Oculta una imagen deseada en la escena.

Método de envío: **POST**.

Parámetros:

String side: posición de la imagen a ocultar. Estas pueden ser: left (izquierda), right (derecha) o both (ambas). (Si este campo es omitido entonces se ocultan todas las imágenes de la escena).

Biblioteca de texturas

Cabeza:

Raza (piel):

- Afro
- Asiático
- Caucásico
- Latino
- Mulata
- Nórdico

Ojos:

- Aceituna
- Avellana
- Azules
- Celestes
- Grises
- Marrones
- Miel
- Negros
- Verdes

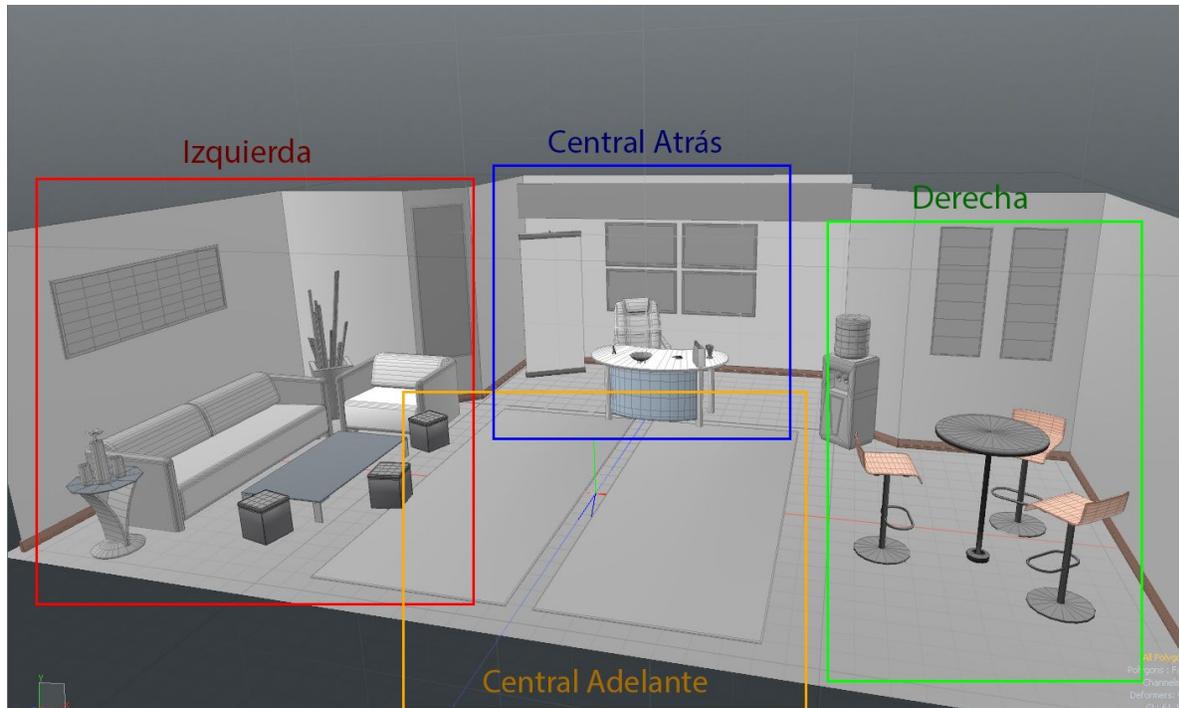
Pelo (uno largo y tres cortos):

- Blanco
- California
- Castaño
- Morocho
- Pelirrojo
- Platinado
- Rubio



Escenario T2Video

Consta de 4 escenarios fijos donde los objetos varían según el ámbito donde se encontrará Clarita.



Centro atrás:

- Podio con Notebook y Taza (izquierda y derecha)
- Rollup
- TVs (4, 2 o 1)
- Silla de escritorio
- Escritorio con escultura/adorno, post-it, reloj, plato/adorno, papeles.





Izquierda y derecha: deben ser ambientes simétricos, tienen los mismos objetos ubicados simétricamente con respecto al centro del escenario.

- Maceta con bambú
- Dispensador de agua
- 3 sofá
- 2 mesitas con florero
- Mesa ratona con notebook, taza y platito
- Asiento cubo
- Lámpara de pie
- Mesa de vidrio
- Sillón
- Cuadro horizontal
- Cuadros verticales
- Sillas altas

Derecha:





Izquierda:



Centro adelante:

- Alfombras verticales
- Alfombras horizontales



Ventana de configuración

T2Video SDK

Autenticación

Client secret

Character ID

Solicitar contraseña para editar ajustes

Configuración

AssetBlundles

Estados de espera

Nombre Workspace

Opciones

Posición cámara

Posición personaje

Posición central

La ventana de configuración permite modificar los datos de autenticación y configuración del programa. Para activarla presionar: alt (izq) + C.

Autenticación

Los campos de autenticación autorizan o deniegan el acceso a las funciones del programa.

Client secret

El client secret es la clave del cliente, necesaria para obtener los servicios proporcionados por la Cloud T2Company tales como Speak, AskWatson, etc. En caso de tener asignado un valor inválido aparecerá un mensaje de alarma advirtiendo de la inhabilitación a los servicios.

Character ID

Character ID es el identificador representante del personaje a utilizar. Se compone por un character (Clara, Xuxa, Ema) y un identificador (por ejemplo: Xuxa 3). Una acción será efectuada únicamente si este campo coincide con el Character ID enviado en las peticiones.

Solicitar contraseña para editar ajustes

Exige la contraseña del cliente para volver a abrir la ventana de ajustes.

Configuración

Los campos de configuración permiten determinar características que el programa tomará. AssetBundles: son paquetes de modelos, texturas, animaciones, etc, proporcionados por el equipo de T2Company. Son aplicables al programa, tanto para enriquecer los elementos ya existentes como para agregar nuevos,

Workspaces: son sitios desde donde definir las respuestas que dará el personaje a las preguntas realizadas a través de la operación AskWatson o la interfaz de escritura y micrófono.

Más info:

<https://help.workspace.ibm.com/hc/es/articles/230038427--Qu%C3%A9-es-Watson-Workspace->

AssetBundles

Este campo contiene el botón “*Editar configuración*” que al clicarse despliega en pantalla una nueva ventana donde se definen cuales AssetBundles aplicar, la carpeta que los contiene y cual archivo define qué objetos aparecen en el escenario.

Estados de espera

Los estados de animaciones de espera están predefinidos, son un conjunto de animaciones que hará el personaje durante los momentos de espera (cuando no se le esté solicitando acciones). Las mismas son elegidas de manera aleatoria y ejecutadas en un período de tiempo también aleatorio. De no optar por ninguno se tomará por defecto PayingAttention.

Talking	No realiza animaciones de espera.
PayingAttention	Tose, estornuda y asiente.
NotPayingAttention	Realiza un gran número de animaciones de espera.

Posición cámara

Permite cambiar la distancia de la cámara al personaje. Al igual que las posiciones del personaje, éstas están predefinidas por el equipo artístico de T2Video.

Posición personaje

Despliega opciones de posiciones donde ubicar al personaje que está en escena. Las mismas están preestablecidas para el escenario a utilizarse.

Posición central

Muestra opciones de posiciones donde ubicar al personaje que está en escena, a la distancia “Centro lejos” y sin mover la cámara.

Nombre Workspace

Este campo contiene una lista desplegable con los nombre-lenguaje de los distintos Workspaces ya insertados. Utiliza todos los Workspaces cargados del idioma correspondiente.

Agregar (+)

Este botón abre una nueva ventana que permite insertar un nuevo Workspace.

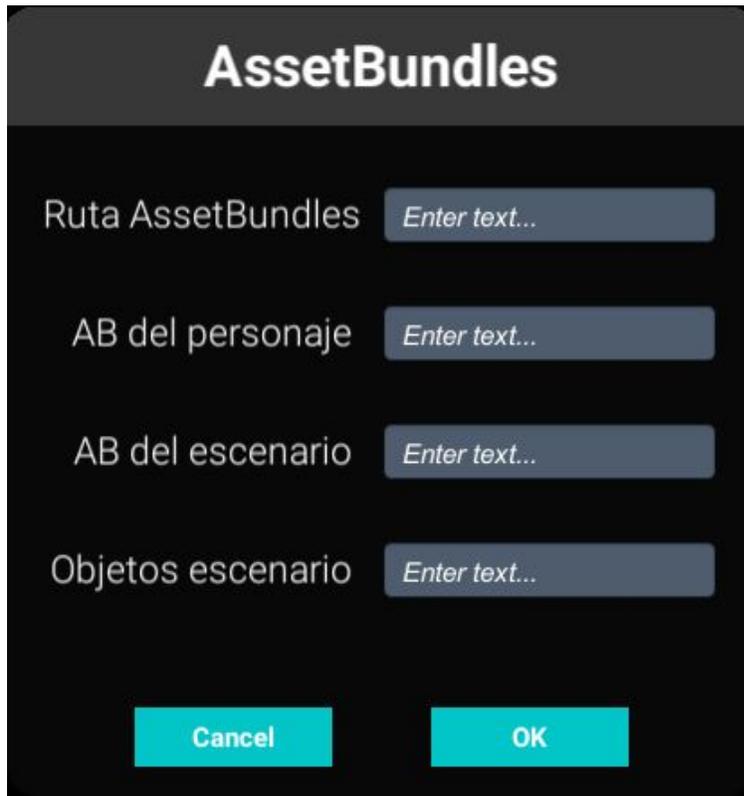
Remove (-)

Al clicar en este botón se remueve el Workspace seleccionado en el campo “Workspaces name”.

Editar

Abre una ventana con datos del Workspace seleccionado en “Workspace name” para verlos y modificarlos.

Ventana de elección de AssetBundles



The image shows a dialog box titled "AssetBundles" with a dark background. It contains four text input fields, each with a placeholder "Enter text...". The fields are labeled "Ruta AssetBundles", "AB del personaje", "AB del escenario", and "Objetos escenario". At the bottom, there are two buttons: "Cancel" and "OK".

Ruta AssetBundles

Es el “camino” o ruta de acceso a la carpeta desde donde se obtendrán los AssetBundles. De estar vacío se utilizará una ruta por defecto.

AssetBundle del personaje

Es el nombre del AssetBundle (ubicado en *AssetBundles path*) que será aplicado al personaje de la escena. Este referencia a un AssetBundle que aporta texturas y modelos de vestimenta, pelo.

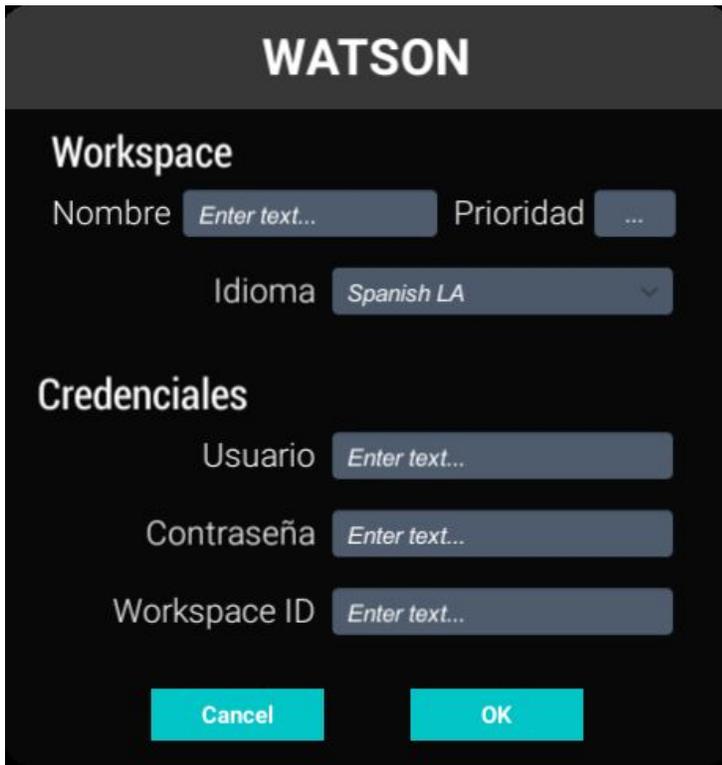
AssetBundle del escenario

Es el nombre del AssetBundle (ubicado en *AssetBundles path*) que será aplicado al escenario de la escena. Este referencia a un AssetBundle que aporta objetos y texturas.

Objetos escenario

Contiene el nombre del archivo de texto con estructura de JSON (ubicado en *AssetBundles path*) que define los objetos del escenario que se encuentran encendidos y apagados.

Ventana de creación y edición de Workspaces



The image shows a dark-themed dialog box titled "WATSON". It is divided into two main sections: "Workspace" and "Credenciales".

Workspace section:

- Nombre:** A text input field with the placeholder "Enter text...".
- Prioridad:** A dropdown menu with a three-dot icon.
- Idioma:** A dropdown menu currently showing "Spanish LA".

Credenciales section:

- Usuario:** A text input field with the placeholder "Enter text...".
- Contraseña:** A text input field with the placeholder "Enter text...".
- Workspace ID:** A text input field with the placeholder "Enter text...".

At the bottom of the dialog, there are two buttons: "Cancel" and "OK", both in a light blue color.

Workspaces

Nombre

Este campo lleva el nombre que junto al lenguaje son el identificador del Workspace.

Prioridad

Aquí se da un valor natural de prioridad al Workspace, esta prioridad servirá para definir a qué Workspace de la lista preguntar antes, cuanto menor sea el valor mayor prioridad obtendrá.

Idioma

Determina el lenguaje en el cual se le harán preguntas al Workspace, este debe ser el mismo que el configurado en la página de IBM.

Credenciales

Usuario

Es el usuario de acceso a la cuenta de IBM para creación y edición de Workspaces.

Contraseña

Contiene la contraseña de acceso a la cuenta de User.

Workspace ID

Este lleva la clave "Workspace ID" del Workspace obtenida en detalles del mismo en la página de IBM.