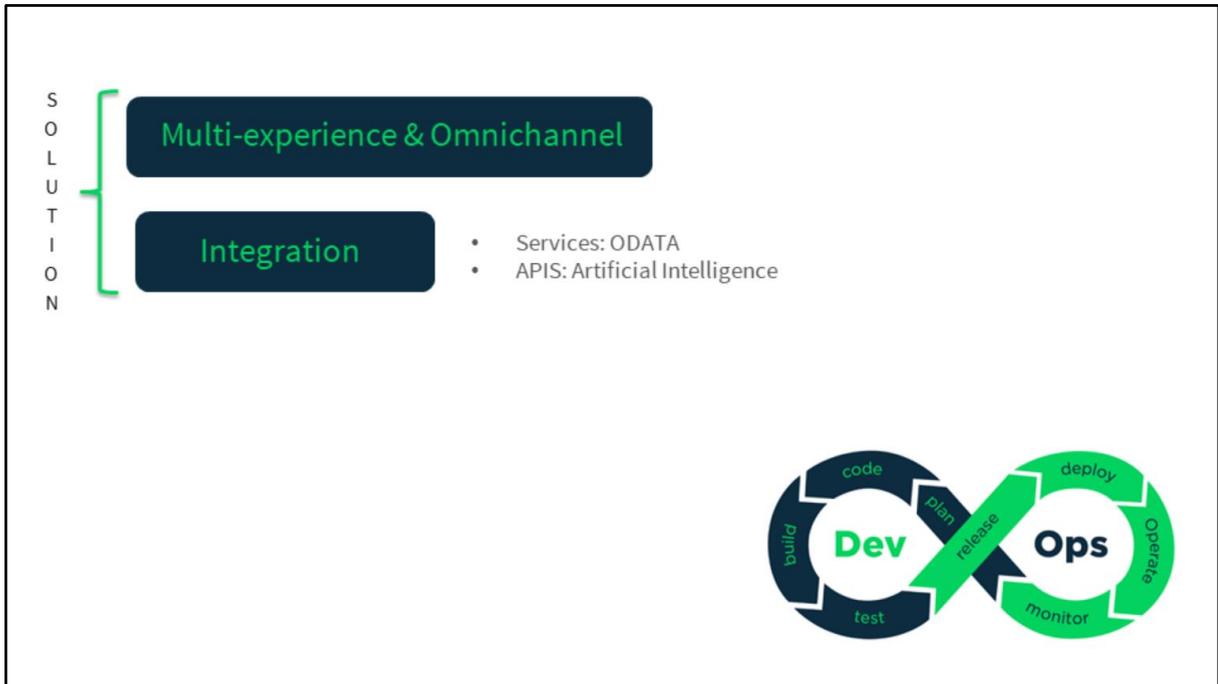


GeneXus[™]
The power of doing

Integration

GeneXus™ 16



Como bien dijo Cecilia, vamos a estar hablando del tema de Integración.
Yo en particular voy a estar hablando sobre el tema de integrar ODATA en proyectos GeneXus.



GeneXus

Quizás algunos ya lo sepan, pero para los que no, ODATA es un protocolo REST que decide estándares de web como HTTP, Atom/XML o JSON.
Este fue desarrollado por Microsoft en el 2007, ya cuenta con una versión 4, es una tecnología flexible que permite la interoperabilidad entre data sources, aplicaciones y clientes.

Services

SOAP

OPEN API (Rest)

+

Services through Data Stores

ODATA

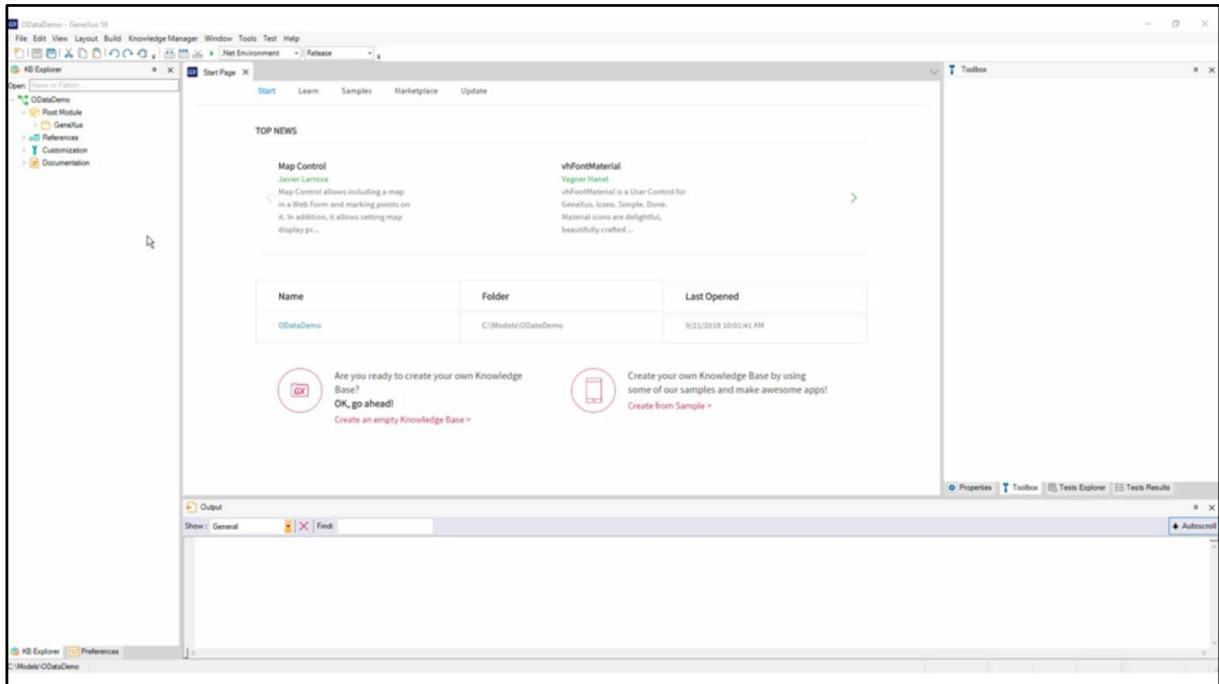
Data View → Transaction → Attributes → for each

¿Por qué incorporar ODATA a sus proyectos GeneXus?

Porque cuando se consume el servicio ODATA, GeneXus toma las identidades que expone este servicio como transacciones, lo cual facilita mucho a los desarrolladores en el tema de el uso de los datos, porque se puede recorrer como un for eachs, se puede aplicar el patterns en las transacciones importadas, se puede aplicar las reglas de negocio a las transacciones y se pueden por ejemplo, insertarlas en un Web Panel.
También los datos de la conexión al servicio se almacenan en un... "como si fuera un Data Store más"



Y ahora vemos una demo de como poder trabajar con Odata desde GeneXus.



En este ejemplo estaremos consumiendo Odata desde el servicio TripPin

Para consumir Odata desde GeneXus debemos ir a: Tools – Application Integration – External DataStore Service Import

Una vez aquí, se nos abrirá una nueva ventana donde ingresaremos todos los datos necesarios para la conexión con el servicio.

Primero nos crearemos un DataStore de tipo Service al que llamaremos: Service TripPin; Este DataStore será el encargado de almacenar toda la información relacionada a la conexión con el servicio.

Luego ingresaremos la dirección del servicio el cual se quiere consumir, en este caso TripPin. Los campos User y Password se utilizan con el servicio “requiere autenticación”, en este caso no es necesario.

En el campo Connection Info, ingresaremos información adicional para la conexión, en este caso indicamos que el servicio no soporta comparadores del tipo Odata **Stream(?)**, hacemos click en “Inspect”, se nos abrirá una nueva ventana en donde veremos las entidades a importar, por defecto GeneXus muestra las entidades principales.

Desmarcando la opción “Show only top level” nos muestra todas las entidades que expone el servicio.

En este caso, trabajaremos con las entidades principales. GeneXus se encargará de importar las entidades adicionales que se necesitan para trabajar con estas.

Una vez importado, vemos que a las entidades GeneXus las importa como transacciones y Data View, por ejemplo, vamos a la transacción “ Airlines” y le aplicaremos el “Patterns work with”, salvamos.

Luego crearemos una Procedure “ChangeAirlinesName” que recorra los datos de Airlines y lo modifique... Para eso usaremos un for each con recorrer Airlines y si el código de la aerolínea empieza a con “A”, le agrega el prefijo “Airline”, salvamos.

Luego crearemos un Web Panel “AirlineChanges” desde donde llamaremos a la procedure. Al Web Panel le agregaremos un botón “AirlinesChangesName”, luego vamos al evento asociado al botón y llamamos al procedure “ChangesAirlinesName” y a “WorkWithAirlines” (WWAirlines), salvamos y ejecutamos.

Vamos a Work With Airlines (WWAirlines) donde insertaremos un nuevo registro de código "AU" y nombre "Airlines Uruguay.

Volvemos al Developer Menu, vamos al Web Panel que creamos (Airline Changes) y presionamos el botón para que ejecute el procedure.

Una vez terminado de ejecutar el procedure, nos lleva a Work With Airlines, donde podemos observar los cambios en el nombre de las aerolíneas.

Por último, vamos a borrar el registro que creamos.

Eso fue una simple demostración básica de como poder trabajar con un servicio Odata en GeneXus, como vieron, se importan las entidades que se importan como transacciones y los datos se trabajan simplemente como si fueran atributos normales de transacciones, se **recorre(?)** un for each, igual que para el patterns, se puede **instanciar(?)** en un Web Panel. Esto mejora abundantemente la producción porque le saca el problema a los desarrolladores de tener que estar preocupándose de escribir todos los "post" y "get" para tener en cada transacción importada a mano desde el servicio.

Y con esto sería bastante, como trabajar con Odata.

- Hola, el tema de la performance de esto comparado esto con trabajar con los motores tradicionales ¿esto es mucho más lento, es mas o menos igual? ¿cómo será?
- Tenemos que considerar que esto está trayendo los datos de un servicio que es externo, esta en Internet, o sea, hay que considerar que en los locales, los DNS locales están todos en máquina, entonces los datos los recepciona mas rápido pero al tener conexión a Internet, tienen un ligero cambio de velocidad porque tiene que ir a hacer los post, los get, los update...
- ¿Es uno por cada registro que hay que hacerlo, no?
- No, no...
- No, las consultas en el protocolo Odata ya define, puede hacer paginado, puedo traer "n" registros a la vez, etc. O sea, está el coste igual del Round Trip, la latencia esa la tenés pero después trae por ejemplo cuando haces un for each, trae de a páginas de registros.
- Hola, no... quizá un poco la aclaración... el tema de elegir o no el protocolo Odata no es evaluando performance o ese tipo de cosas, sino, es porque algunos proveedores, en particular "SAP" como capa de interface en alguno de sus productos expone sus entidades utilizando este protocolo, entonces un poco la necesidad de utilizarlo es, para poder integrarme con terceros o mis sistemas disponibilizarlos a terceros cuando nos piden que usemos este protocolo.
- ¿Qué diferencia hay entre... si lo hiciera localmente, hacer una utilización de un Data View vs el Odata?
- La diferencia es que con el Data View tendrías que estar haciendo uno por cada transacción, por cada identidad mejor dicho... con el Odata ya GeneXus te hace la inspección del servicio y te trae todas las entidades como transacciones y sus respectivos Data Views.
- Disculpá que me voy a meter un segundito, ¿Tu estás queriendo comprar entre hacer un post a mano, hacer los post y los get a mano contra importar el servicio? ¿Esa es la comparación?
- Emm... Si yo tengo un Data View, yo le digo... hago un for each y actualizo información, si es que hago Data, hago un for each y hago la actualización...
- Exacto! Desde el punto de vista de GeneXus, ahora haces lo mismo, o sea... queda integrado el modelo de datos de Odata al modelo de datos de GeneXus... entonces por eso puedo usar for each, etc. Si uno lo hiciera a mano, haciendo post y get, GeneXus no tendría información sobre el modelo de datos que estas accediendo , entonces por ejemplo, en particular no podrías una **"foreign key" (?)** a una entidad Odata y ese tipo de cosas y además, una vez que está integrado el modelo Odata al modelo GeneXus, por ejemplo las navegaciones que resuelve GeneXus aplican también al modelo de Odata.
- ...Si, yo quería preguntar por el tema de SAP, me interesó la comparación de Odata y el

- "conector" digamos... ¿Qué es más fácil en el mundo SAP hacer?
- A ver... son protocolos totalmente distintos, "RFC" es mas nativo y quizá es más rápido que esto y más directo cuando estas dentro del área de la "institución" digamos, esto lo podés poner más para afuera ¿qué te conviene hacer? no te puedo recomendar...
 - No, no! No es lo que me conviene a mi... yo preguntaba por lo que... el cliente que tiene SAP prefiere hacer, digamos... si exponer las "**bapi**" o exponer Odata...
 - Más fácil es las "**bapi**" porque del lado de SAP no hay que hacer nada, están accesibles, para vos cuando utilizas Odata contra ese "**AP**" para el "**R3**" digamos... atrás están las mismas "bapis", lo que vos tenés que hacer es... elegir cuales querés utilizar con el protocolo Odata y publicarlas como servicio Odata, entonces hay una especie de trabajo adicional desde el lado de SAP.
 - ¿ Y es fácil para ellos hacer eso?
 - Es fácil para ellos hacer eso, pero a vos no te conviene porque "dependes" de ellos digamos, ¿no? O sea, dependes de una "**baper**" dependes de alguien de infraestructura de ellos de que les tenés que pedir que te publiquen tales bapi como Odata, entonces te queda más como acceder directamente al **repositorio** que ahí no tenés que hacer nada ni dependes de nadie y además la llamada a "RFC" es mas performante... eso en cuanto al "R3", esto más que nada apunta a la otra versión de ese "ap" que es Bussines One que ahí no tenemos bapis, entonces Bussines One es el único camino que tendríamos para hacer una integración mas redonda ¿no?
 - Estamos consumiendo, pero ¿tenemos posibilidad de producir servicios Odata desde Genexus?
 - De momento GeneXus no provee Odata, estamos trabajando en eso, en un update siguiente vamos a liberarlo.

Artificial Intelligence

Artificial Intelligence

IBM Watson

Microsoft Cognitive Services

SAP Leonardo

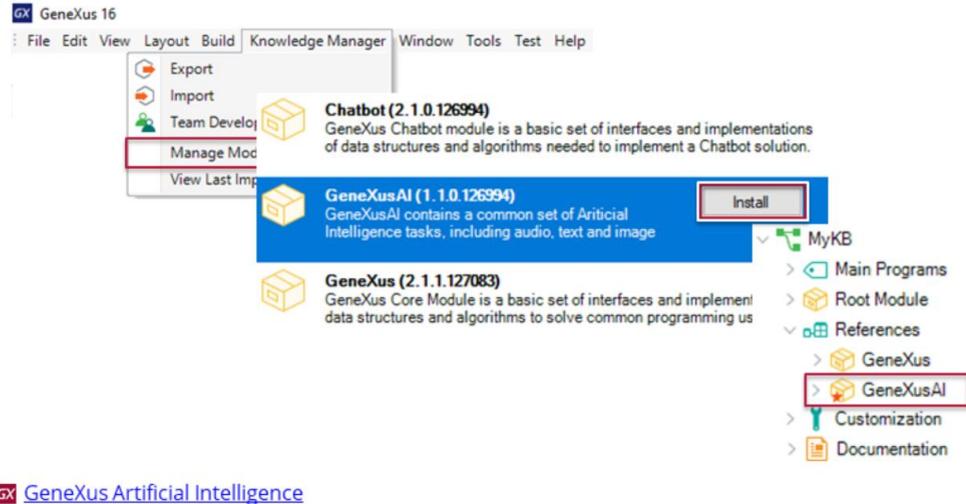
Google Cloud Services

Amazon Web Services

En esta segunda parte de Integración, lo que yo les quería comentar es que estamos trabajando la instalación de varios proveedores en el Cloud sobre Inteligencia Artificial, en particular ahora en GeneXus 16, estamos integrando lo que es: IBM Watson, Microsoft Cognitive Services y SAP Leonardo.

En próximos upgrades vamos a estar liberando también lo que es Google Cloud Services, todo lo que tiene que ver en esto de inteligencia artificial, después también lo que es Amazon Web Services y bueno, vamos a ir integrando, vamos a tratar de hacer también, cosas más orientadas a la parte Offline, a trabajar con (...) con estos modelos que no tienen porqué consumir algo desde la nube.

Artificial Intelligence



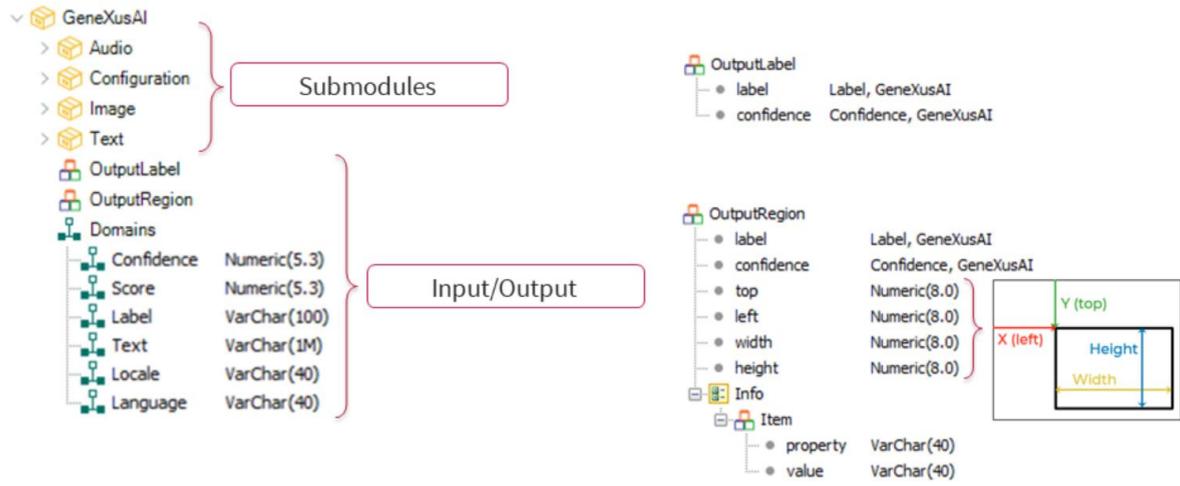
Bien... ¿En qué estamos trabajando?

Estamos trabajando en una API ¿Sí? Sobre todo estos proveedores, para no tener que aprender soluciones particulares, no tener que aprender la solución de Microsoft o aprender la solución de SAP, y lo que estamos ofreciendo ahora es un módulo **"Build Teem" (?)** dentro de GeneXus que simplemente lo instalamos yendo por: > KnowledgeManager > Manage Module References. Ahí se van a listar todos los módulos que es donde estamos ofreciendo Build Teem, entre ellos va a estar GeneXus AI, entonces simplemente le dan al botón "Install" y ahí ya lo van a tener integrado en su base de conocimiento ¿sí?

Dentro del nodo "References" donde está también el GeneXus core, donde ustedes tienen en cuenta los **"External Object"** y otras cosas que ya viene predefinidas.

Algo que les quiero comentar es que, todo esto está todo en el Wiki, hay una TOC referente al tema que ustedes pueden leer ahí, esta bastante ya... bastante completa y vamos a ir trabajando también sobre eso.

Artificial Intelligence



Bueno... ¿Cómo está estructurado este módulo?

Este módulo tiene cuatro sub-módulos como ven ahí y ahora vamos a empezar a detallar que hace cada uno de ellos, y después una serie de dominios y tipo de datos que nos van a ayudar para la entrada/salida de los métodos que vamos a hablar ahora a continuación.

Por comentarlos un poquito nada más ¿no? Los nombres son bastante intuitivos...

- Confidence: para el dominio Confidence sería un nivel de confianza que el proveedor nos devuelve a partir de una predicción que haya hecho.
- Score: sería una puntuación, en este caso estamos tratando entre 0 y 1, la semántica de ese puntaje depende del método que estemos usando, que ahora después se lo voy a explicar cuando vayamos escribiendo los métodos.
- Label: es simplemente una etiqueta
- Text: es texto
- Locale y Language: bueno, "Locale" es algo más particular que el lenguaje, el lenguaje es algo más general.

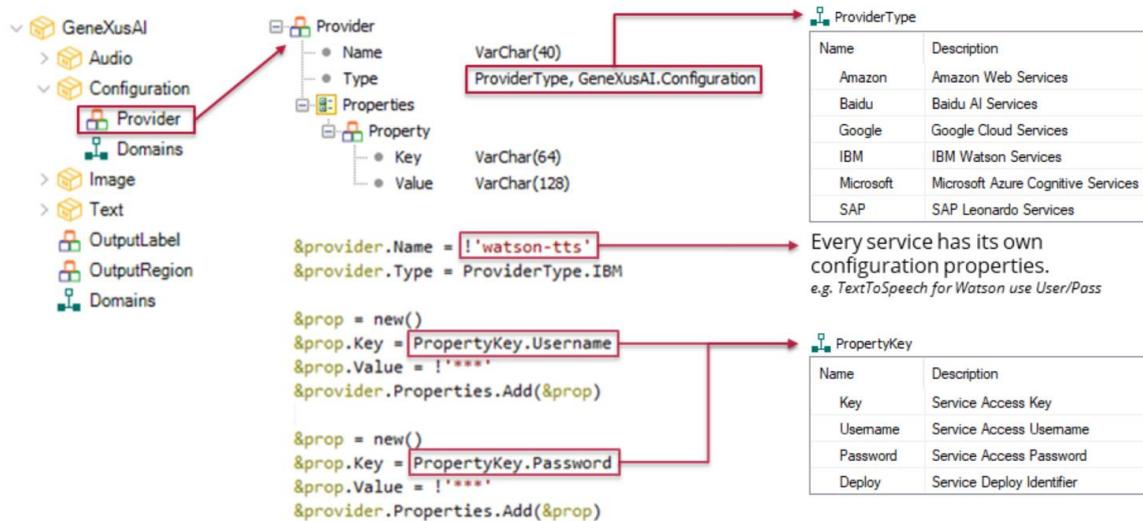
"Language" es por ejemplo: el español de España vs el español de México, o el Inglés de Estados Unidos vs el Inglés de Inglaterra. Todo eso lo vamos a ver un poco más después, cuando hablemos un poco más de los métodos.

Los tipos de datos

Primero el que teníamos allá es el "OutputLabel" que simplemente es una estructura que tiene dos cosas combinadas de aquellos que es "la etiqueta" y una "confianza", o sea, cuando el modelo nos devuelve una etiqueta y que tan confiable... que confianza tiene sobre esa etiqueta, lo que nos da el modelo y por el otro lado "OutputRegion" que nos va a permitir... tiene también las dos etiquetas "Label" y "Confidence", pero además lo que nos permite... nos está dando una región. Una región sobre lo que nosotros le estemos pasando de entrada, esto sirve para imágenes, como vamos a ver a continuación y básicamente lo que nos da, es: "Top", "Left", "Width" and "Height" y.. ¿Qué es eso? Bueno, justamente para poder dibujar un rectángulo, el rectángulo que vemos aquí... definimos lo que es el "top", que sería el "left", "height" y "width", con esta información ya podemos dibujar este rectángulo sobre una imagen, por ejemplo.

Bueno y esto simplemente es metadata, que podría llegar a ver, es información adicional que nos devuelve el proveedor.

Artificial Intelligence



Bueno, ahora si...

El primero módulo que vamos a comentar es el "Configuration", este es un módulo muy importante porque es común a todos, donde tenemos ese SDT que es el mas importante, donde nosotros vamos a configurar las propiedades para nuestro proveedor, mediante un nombre que va a ser por lo general el nombre del servicio que estemos utilizando... el "tipo de proveedor" y después una serie de propiedades que nosotros vamos a ir configurando. ¿Qué tipo de proveedor? Bueno, eso es un domino enumerado donde vamos a tener todos los proveedores que vayamos a ir soportando. Por el momento como les comenté, tenemos "Microsoft", IBM y SAP, estos tres últimos. Vamos a ir incorporando otros mas adelante.

Y bueno, ¿Cómo programamos esto?

Como les decía hoy: tenemos una variable de tipo Provider, seteamos un nombre, en este caso estamos diciendo que vamos a usar el "TextToSpeech", esa es la sigla de Watson.

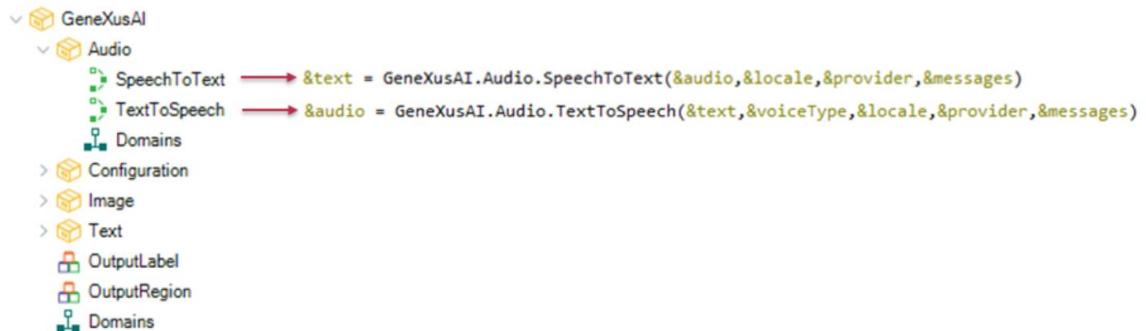
ProviderType: en este caso IBM.

Y esto es algo muy importante. Que cada servicio tiene su propia configuración de propiedades, por ejemplo: el "TextToSpeech" requiere utilizar un Username y Password para acceder al servicio y eso es justamente lo que hacemos luego, creamos una propiedad, le decimos que es el Username, le decimos que nuestro valor es... cual es el username que nos esta dando el servicio, lo agregamos al conjunto de propiedades para ese proveedor y también agregamos la Password.

Con esa configuración ya tenemos todo.

El dominio este: "PropertyKey" nos va a ayudar a decidir cuales son las propiedades que tenemos disponibles, por el momento tenemos "Key" cuando le servicio se basa en una API key y Username/Password que por lo general se ponen en simultaneo... y mas adelante vamos a estar viendo algo de lo que es un identificador de Deploy cuando tengamos modelos Custom, uno de los modelos default que proveen cada proveedor.

Artificial Intelligence



Bien, ahora si... la parte interesante

Primero módulo Audio, el módulo Audio cuenta con dos procedures, el primero es: **SpeechToText**, que lo que hace es... tengo un audio, le digo también cual es "locale" en el que estaría ese audio Acuérdense que es el lenguaje específico, el español de España, por ejemplo, le pasamos el SDT provider ya establecido y esto es simplemente para manejar lo que son los errores que nos podría llegar a devolver el sistema, si nos olvidamos de algún parámetro, alguna configuración o algo, eso va a venir acá.

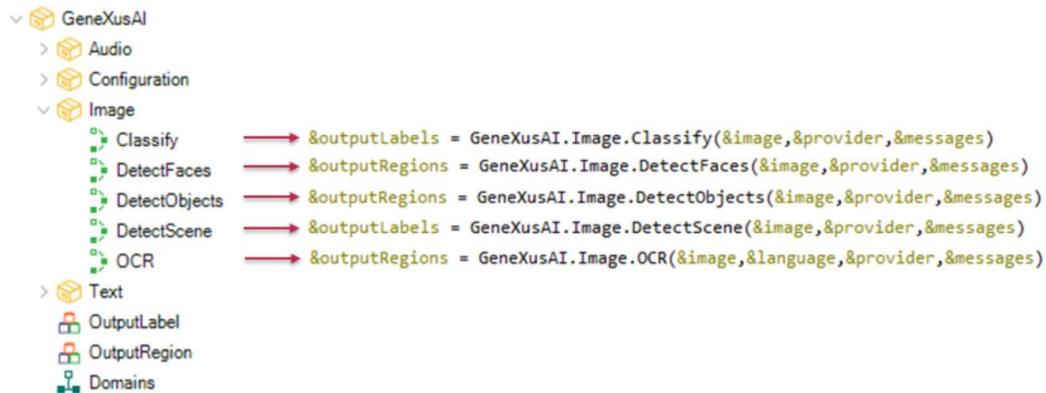
Bueno, lo interesante es que a partir de ese Audio, nos da el texto plano de lo que se esté hablando ahí.

De forma análoga tenemos el **TextToSpeech**, que es al revés, dado el texto... nos devuelve el audio.

Y tenemos algunos parámetros adicionales, como por ejemplo, que tipo de voz queremos, si queremos que sea una voz femenina o masculina, en lo que es lo mismo.

Y en estos últimos dos parámetros siempre son comunes a todos los métodos que vayamos a encontrar en GeneXus AI.

Artificial Intelligence



Ahora el módulo Image

Tenemos métodos como: Classify, que dada una imagen, nos va a devolver una serie de etiquetas y su probabilidad, por eso es "OutputLabels" ¿sí? ¿se acuerdan? Habíamos visto el SDT, que OutputLabels tenía dos campos, el primero era la etiqueta en sí mismo y el segundo era la confianza y está en plural porque nos devuelve una conexión de eso, dado una imagen nos devuelve una conexión de etiquetas con probabilidad de lo que pueda llegar a identificar allí, como les comentaba, también vieron que siempre se mantienen estos dos últimos parámetros ... que cumplen la misma funcionalidad que en todos lados.

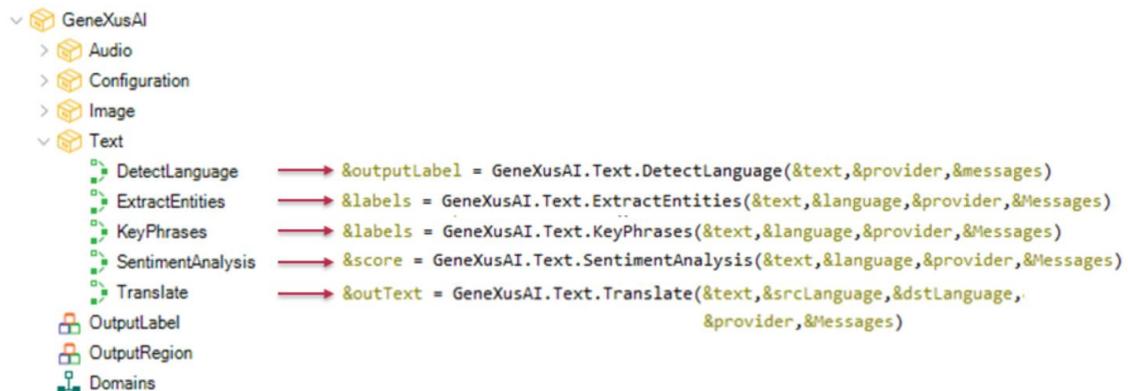
Después tenemos: DetectFaces, que dada una imagen nos reconoce rostros en esa imagen y nos devuelve las regiones de esas caras con el SDT que habíamos comentado anteriormente, DetectObjects es totalmente análogo, también dado una imagen nos devuelve las regiones donde están esos objetos.

DetectScene, que dada una imagen nos devuelve una secuencia de etiquetas que refieren a lo que es, por ejemplo... porque es medio complicado de distinguir éste del Classify... el DetectScene nos detecta en donde estaríamos en una imagen, por ejemplo, si yo me saco una foto hoy en la torre Eiffel, es esperable que en OutputLabels aparezca que estás en la Torre Eiffel, estás en París, estas en... ¿me explico? Detect a diferencia del Classify, el Classify nos devuelve que cosas aparecen en esa foto.

Después tenemos OCR, que para el que no conoce la sigla, es una sigla que significa Optical Character Recognition, que lo que hace es, a partir de una imagen, nos devuelve el texto que aparece en esa imagen o varios textos... también podemos indicar un lenguaje en el que está escrito ese texto y los parámetros comunes y nos devuelve donde están indicados esos textos.

Y bueno... con eso terminamos el módulo Image.

Artificial Intelligence



Y ahora yendo al último...

También tenemos primitivas para detectar el lenguaje, que dado un texto nos devuelva una única etiqueta si este está en singular y no en plural, nos devuelve un lenguaje y una confianza...

-perdóneme! Volviendo a lo anterior, al OCR ¿dijiste que devuelven las regiones? ¿no devuelve el texto?

- Perdón... sí, devuelve el texto, pero acordate que el SDT OutputRegion tiene, una etiqueta, que ahí vendría a tener una palabra, por ejemplo, un Confidence (o sea, la confianza de que esa palabra sea esa) y además, donde está ubicada en la imagen esa palabra que encontré.
- Ok
- Eso era lo que decíamos hoy del "top", "left", "width" "hight"...
- ¿Y eso dónde lo esta devolviendo?
- Eso lo devuelve todo en este SDT
- Ah, perfecto!
- Igual ahora vamos a ver una Demo justo de este caso, así que capaz que ahí te queda más claro

Bien, estábamos acá en DetectLanguages, "ExtractEntities", nos da un texto, nos dice que entidades aparecen en ese texto, entidades entendidas como un nombre propio, el nombre de una organización, el nombre de un país, cosas que puede mostrar ese texto y nos devuelve una secuencia de etiquetas, esas etiquetas corresponden con esos nombres.

KeyPhrases, a partir de un texto nos devuelve lo que son frases claves e incluso palabras claves. Porque será como un conjunto más acotado, por eso también es una colección de etiquetas, cada etiqueta representa una frase.

"Análisis de sentimientos", que dado un texto nos dice si una opinión que está en este texto es positiva o negativa... y acá lo que les comentaba hoy del "Score", este puntaje va entre 0 y 1, en este caso el 0...

- Me saltó una duda... ¿todos los "resultados" que obtengo van a depender del proveedor, verdad?

- Exactamente!
- ¿Cada uno de estos puede tener un proveedor distinto? ¿Por qué un proveedor puede ser mas fuerte... en texto, el otro en imágenes...
- Por supuesto, puedes configurar.. Utilizar SentimentAnlysis para Microsoft, utilizar DetectLanguages para IBM, por ejemplo porque capaz que te convence más, lo único que haces es configurar el proveedor para ese caso.
- Correcto... ¿y tu puedes hacerlo en la misma (KB ?) sin problemas? ¿Pueden convivir?
- Si, por supuesto... es como codifiques esto, cuando envías este parámetro vos decís si lo querés por Microsoft, por IBM y no importa el método que uses, podés hacerlos intercalados.
- Ok, gracias!
- Por nada.

Bien... el tema del puntaje... este era un puntaje entre 0 y 1, que mientras esté más cerca del cero va a ser negativo,... mientras más cerca de 1, positivo...

Y por último, bueno, el típico "Translate" ¿no? El de traducir de un idioma a otro, que dado un texto, un lenguaje, un idioma de origen y un idioma de destino, nos devuelve el texto traducido ¿sí? Tan simple como eso, lo mismo Provider/Messages, lo configuramos como queremos y se acabó la historia.



Proof of Concept

Ahora si... la prueba de concepto que va mas que nada a tu pregunta...

Artificial Intelligence

Control Info	
Control Type	SD Image Map
Auto Grow	False
Image	GXAI_Cam_Placeholder
Horizontal Coordinate Attribute	@outputSquares
Horizontal Coordinate Field Specifier	Item(0).Left
Vertical Coordinate Attribute	@outputSquares
Vertical Coordinate Field Specifier	Item(0).top
Size Attribute	@outputSquares
Size Field Specifier	Item(0).size

Event 'DetectFaces'

```

Composite
  if NOT &image.IsEmpty()
    &outputRegions = DetectFaces(&image,&provider,&Messages)
    do 'Reload'
  else
    msg("Nothing to recognize faces")
  endif
EndComposite
Event
  
```

```

Sub 'Reload'
  Grid1.SetBackgroundImage(&image)
  &outputSquares = SquareRegions(&outputRegions)
  Grid1.Refresh()
EndSub
  
```

Event 'OCR'

```

Composite
  if NOT &image.IsEmpty()
    &outputRegions = OCR(&image,&language,&provider,&Messages)
    do 'Reload'
  else
    msg("Nothing to recognize characters")
  endif
EndComposite
Event
  
```

Primero se los voy a mostrar acá:

Ahí pueden ir viendo como vamos a seleccionar una imagen de la galería, en este caso es un rostro...

ese primer botón es para hacer la detección de caras y podemos llegar a lograr eso ¿sí?

Dibujar esa región allí... ahí simplemente estoy cambiando la metadata que me trajo el servidor, a mi antojo nomás...

Y ahora sí, el segundo caso, tenemos ahí un Broadway, aplicamos OCR y eso es lo que yo les comentaba hoy, acá ¿sí? Que me va a devolver la región e incluso el texto que me identificó.

Bien... ¿Cómo está diseñado este panel?

Simplemente tenemos un (cumb as?), los dos botones, por eso parecen como flotantes si se quiere ahí y tenemos un Grid ¿sí?

Este grid tiene una particularidad que es un SD Image Map, no se cuánto ustedes lo usaron, pero ¿de qué me sirvió esto en SD? Bueno, justamente porque puedo dibujar la esquina superior, la esquina izquierda, o sea el X... donde está la imagen y un size(?), en este caso solo me permite dibujar cuadrados y no rectángulos, pero bueno, es algo que vamos a mejorar próximamente.

¿Cómo hace el Event asociado a este botón?

Bueno, simplemente cuando la imagen no es vacía llamamos al método con el proveedor ya "seteado", devuelve las regiones y hacemos un "Reload" de este grid, ahora después les voy a comentar qué es ese Reload ¿sí? Porque tiene que ver justamente con el control y capaz que no es tan para esta charla pero se los voy a comentar igual más adelante.

Y para el caso de OCR lo mismo ¿sí?

Es totalmente análogo, si la imagen no es vacía, aplico OCR sobre esa imagen, el lenguaje ya lo tengo de origen, ya le había seteado como inglés... y lo mismo, hacemos un Reload.

¿Qué es este Reload que es lo que habíamos comentado hoy?

Las particularidades que tiene este grid, es que a mí me permite setear una imagen de fondo a ese grid y es lo que estoy haciendo acá, seteo la imagen que yo le había pasado, seteo esa imagen de fondo del grid y después lo que acá hago es una transformación, acá hago una transformación de lo que son los rectángulos que me devuelven las soluciones estas y los transforma a cuadrados.

Eso es por lo que les comentaba hoy que tenemos esta limitación ahora en SD por este

momento, en web es mucho más flexible, porque ustedes van a poder dibujar rectángulos más fácil ahora con esto que vieron de los Users Control.
Y bueno, por aquí termina... vamos con la pregunta.

- Supongamos que quisiera hacer un Chat chatbot pero con voz, ¿se podría utilizar el SpeechToText?
- Se podría usar perfectamente sin problemas.
- ¿Y eso sería dentro del SD Panel del chat?
- Del chat... capturas antes el texto...
- ¿Cuándo capturo la voz, como sería? ¿eso habría que guardarla en un archivo y pasarle el archivo al SpeechToText?
- Claro! Algo así... en una variable audio,. Si volvemos un poquito para atrás... es una variable audio nomás, vos podrías hacer... si vos estas en SD, por ejemplo ¿te acordás que tenés una (...) se llama "Audio Recorder", bueno, haces la grabación de audios, la guardas en esta variable y esta variable es la que le pasas nomás al servicio y ahí ya te devuelve el text
- Bien...
- Entonces con eso ya te alcanza.
- Muchas gracias!
- Hola, si tengo por ejemplo, un modelo que yo creé y ya entrené con imágenes, fases y materiales ¿Cómo configuro ahí? Porque creo que está usando un modelo estándar o algo así...
- Ahora si... por el momento estamos usando los modelos default de cada proveedor. Estamos en planos de incorporar lo que es... lo que pasa que esto, que es lo que vos comentas de los modelos Custom es bastante nuevo en los proveedores, lo están incorporando de a poco, no están todos tampoco, están creo que... Microsoft e IBM sí lo ofrecen, pero los demás no estoy muy seguro; todavía no tenemos un mecanismo para incorporarlo pero ya próximamente, seguramente (...) ya vas a tener algo para incorporar, por eso viste que yo hoy comentaba... acá en las propiedades, decía que íbamos a tener un valor Deploy para que vos le puedas identificar tu identificador ¿sí? De donde esta "deployado" ese modelo que vos entrenaste aparte y en ese caso ya podrás utilizarlo nomás, sin problemas.
- ¿Entonces hoy no se puede entrenar un modelo?
- Ahora con el "realize" que tenemos ahora en V16 sub 0, todavía no.
- Ah, ok!
- Hola, agrego un dato, ésta es la primer versión del módulo de GeneXus AI, la idea es que vamos a seguir trabajando en este tema, vamos a tener la opción de incorporar modelos Custom de los proveedores y vamos a tener la opción de agregar proveedores locales, por ejemplo Cor Mdl o Mdl Kit(?) para que corra en el dispositivo mismo y seguramente tengamos opción de usar "tensorflow" o algún otro lenguaje digamos... estamos dando los primeros pasos en esto, lo que agregamos fue la funcionalidad más básica, pero la idea es seguir avanzando por este camino, no tenemos bien definido todo lo que queremos, por ahora lo que sabemos es que en la línea de lo que ya tenemos, agregar nuevos proveedores, pero sí, la idea es seguir trabajando, seguir incorporando funcionalidades en este tema.

S
O
L
U
T
I
O
N

Multi-experience & Omnichannel

- Powering analytics
- BPM

Integration

- Services: ODATA
- APIs: Artificial Intelligence

Security



GeneXus[™]
The power of doing