Portal Analítico Rápido Conversacional y Efectivo

PARCE

Data Sandbox

Fondo Nacional del Ahorro
Equipo de Analítica

Portal Analítico Rápido Conversacional y Efectivo

PARCE

Andres Mateo Quevedo
amquevedo@fna.gov.co

Daniel Jaramillo Alvaréz
djaramilloa@fna.gov.co

Diana Milena Cuervo Paloma
dcuervo@fna.gov.co

Julian Guillermo Carrillo Meneses
jgcarrillo@fna.gov.co

David Armando Prieto Naranjo
dprieto@fna.gov.co

Miguel Andres Diaz Ezquivel

Fondo Nacional del Ahorro Carlos Lleras Restrepo
Equipo de Analítica
Data Sandbox
MinTIC

Índice

Introducción

Este documento describe la participación del Fondo Nacional del Ahorro (FNA) en el Data Sandbox 2021 del Ministerio de Tecnologías de la Información y las Comunicaciones (MinTic). Es un articulo meramente divulgativo que propende impulsar por medio de la experiencia en el Data Sandbox a que entidades publicas materialicen sus proyectos de Big Data e Inteligencia Artificial con el uso de tecnologías en la nube.

El Portal Analitico, Rápido, Conversacional y Efectivo (PARCE) es el proyecto desarrollado por el FNA en el Data Sandbox, este trabajo propende definir un ambiente conversacional integral en pro de ampliar la experiencia de los usuarios y la ingesta de información que permita el análisis de las interacciones para la toma de decisiones. Además de fortalecer conocimientos en la plataforma de nube de Microsoft Azure para trabajar proyectos inhouse de Big Data e Inteligencia Artificial (IA).

1. Data Sandbox

La plataforma Data Sandbox es un espacio colaborativo del **MinTIC** (Ministerio de Tecnologías de la Información y las Comunicaciones) para las entidades públicas del país, en donde se disponen servicios de *nube* para realizar proyectos *piloto* de **Analítica** y **Big Data**.

La plataforma es empleada para explorar conjuntos de datos de manera colaborativa, con el fin de crear, probar, ensayar y definir soluciones Big Data aplicables a problemáticas públicas/ciudadanas. El Data Sandbox dispone de **plataforma cloud** para los desarrollos, luego, a los proyectos se les habilitan herramientas con altas capacidades para el **almacenamiento** y **procesamiento** de datos estructurados, semiestructurados y no estructurados a través del uso de tecnologías de Big Data.

Esta versión del Data Sandbox como iniciativa de MinTIC inició en el segundo semestre del 2020 y contractualmente va hasta finales del 2021.

1.1. Entidades en el Data Sandbox

Con el propósito de crear y definir nuevas instancias analíticas para desarrollos internos de las entidades, a la fecha se han ejecutado varios proyectos en el Data Sandbox, estas participaciones corresponden a los siguientes proyectos:

- (6) Estadísticas
- (3) Inclusión social
- (3) Desarrollo rural
- (1) Comercio, industria y turismo
- (1) Educación
- (1) Planeación
- (1) Vivienda
- (1) otros sectores

En el portal de Datos Abiertos puede encontrar más información respecto a los proyectos en fase de inicio, en desarrollo y/o finalizados en el Data Sandbox. [3]

1.2. Azure para el Data Sandbox

Una **plataforma de nube** es un conjunto de servicios tecnológicos diseñados para ofrecer características de efectividad e innovación que facilitan el desarrollo de proyectos científicos y analíticos.

Los servicios de nube conforman una insfraestructura completa e integral a disposición de creación de proyectos con elementos claves como:

(1) Almacenamiento de grandes volumenes de datos.

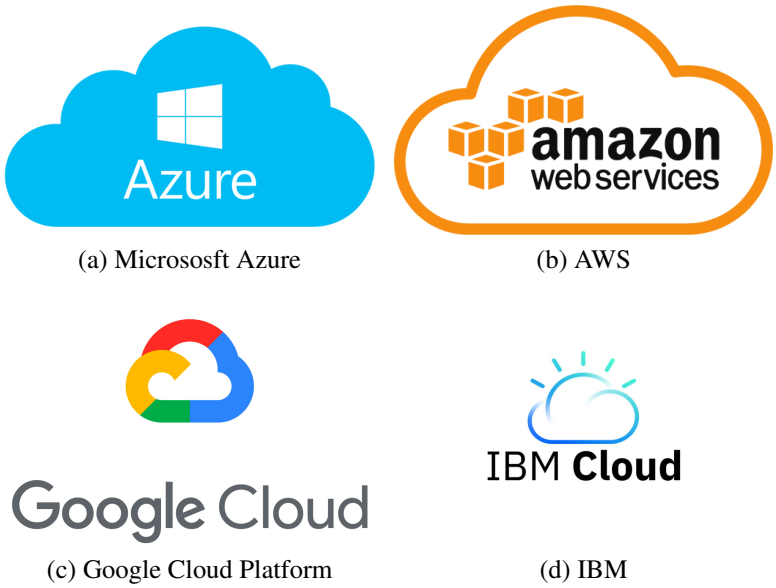
(2) Procesamiento de datos.

(3) Analítica de datos.

(4) **ML**: Machine Learning.

(5) **IA**: Inteligencia Artificial.

Existen varias plataformas cloud en el mercado:



(a) Micrososft Azure (b) AWS

(c) Google Cloud Platform (d) IBM

Figura 1: Plataformas cloud

El Data Sandbox del MinTIC dispone de **Azure** (Plataforma de nube de Microsoft) para el desarrollo de los proyectos de Big data de las entidades públicas.

Azure es la plataforma de nube desarrollada por **Microsoft**, ofrece servicios de computo, almacenamiento de información, diseño, creación y administración de aplicaciones en la nube.

Azure permite un uso básico y gratuito de sus servicios para probar y aplicar sus herramientas a proyectos y desarrollos sencillos, además permite aprender a usar sus servicios de la siguiente forma:

Servicio	Beneficio	Enlace
Azure free account	Cuenta gratuita de Azure: - 12 meses gratuitos de servicios populares. - $200 dólares para explorar Azure.	Azure Free
Microsoft Learn	Rutas de aprendizaje diseñadas para conocer de Azure Cloud.	Azure Learn
Training Days	Conferencias guiadas por expertos de Microsoft, donde podrá obtener vouchers para presentar exámenes de certificación de Microsoft Azure.	Azure Training

Los servicios de Azure se dividen en dos grupos principales:

(1) Servicios de plataforma

(2) y, Servicios de infraestructura

Puede consultarlos en detalle aquí.

En particular, para el desarrollo de **PARCE**, los servicios de mayor interés son:

(1) Data: SQL Database, Azure Synapse Analytics, Cosmos DB.

(2) Intelligence: Cognitive Services, Bot Services.

(3) Analytics: Machine Learning, Data Lake Storage.

2. Data Sandbox: FNA

2.1. Motivación

De ejercicios exploratorios y de modelos descriptivos para definir nuevas estrategias de atención, fuentes de datos complementarias y con el objetivo de apotar a la Transformación Digital del FNA, surgió un voluminoso insumo de datos de información conversacional de usuarios que en algún momento se comunicaron de forma virtual con la Entidad.

A partir de dicha captación de miles de mensajes de usuarios representados como datos de texto, se propuso la siguiente pregunta: ¿Cómo crear nuevas experiencias conversacionales a los usuarios del FNA con base en el conocimiento de registros conversaciones?

La respuesta a la anterior pregunta concluyó en lo siguiente:

(1) Definir técnicas y conceptos para la manipulación de datos que no tienen una estructura clara y definida, que en nuestro caso, son miles de mensajes de usuarios.

(2) Después de procesar los datos semiestructurados, crear datos de entrenamiento para materializar un ambiente conversacional con inteligencia cognitiva e ingesta de datos conversacionales.

Lo anterior definió un Portal Analítico ágil y rápido para desarrollar analítica de datos de información conversacional de los usuarios que buscan comunicarse con el FNA.

2.2. Postulación

Dentro de conversaciones de seguimiento sostenidas con los profesionales de MinTIC, socializamos algunas de las iniciativas trabajadas por los integrantes del Equipo de analítica del FNA que fueron construídas con el objetivo de tomar decisiones basadas en los datos. Modelos de Machine Learning, modelos matemáticos y georreferenciados, son algunas de las aplicaciones conversadas en este espacio. De esta forma es como el FNA evidencia que tiene la capacidad, el talento y la oportunidad de llevar sus preguntas de negocio y los modelos a otro nivel, a la nube mediante el Data Sandbox del MinTIC.

La iniciativa nace con la postulación a la convocatoria 'Big Data al servicio de las entidades públicas' en donde el proyecto PARCE del FNA ocupó el segundo puesto, entre 34 proyectos postulados y sólo fueron seleccionados los 6 primeros para desarrollar sus iniciativas.

3. PARCE

De la interlocución de dos individuos se pueden captar datos relevantes como: Información de contacto, datos de identificación, interéses o preferencias en productos, etc... Ahora, imagine **miles** de individuos (usuarios) interlocutando por medio de un canal de atención, esto produce un volumen increible de información para entender las necesidades de los usuarios.

PARCE está ideado para usar datos captados por canales virtuales, con el objetivo de desarrollar un ambiente de atención y analítica enfocado en la experiencia del usuario.

En este capítulo presentamos el desarrollo de PARCE en el Data Sandbox, desde ideas preliminares hasta el concepto final del proyecto, en específico seguiremos estos pasos:

(1) Conceptos e ideas preliminares de PARCE.

(2) Diseño de la arquitectura del proyecto.

(3) Uso de **DataLake** para almacenamiento de insumos de datos.

(4) Operación de los datos en **DataBricks** para el aprovechamiento de las capacidades de procesamiento de **Spark**.

(5) Exploración de los servicios cognitivos: **LUIS** y **QnA Maker**, para la creación del asistente conversacional de PARCE.

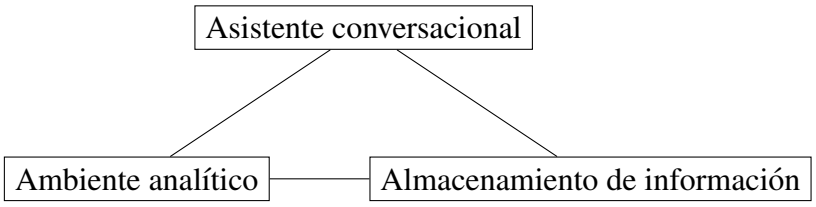
(6) Exploración y uso de **Bot Framework**, con SDK como ambiente de desarrollo del Bot de PARCE.

(7) **Cosmos DB** para guardar información de las conversaciones del ChatBot.

(8) Consumo de los datos desde **Cosmos DB** hacia el tablero de visualización de PARCE desarrollado con **Azure Synapse Analytics** y **Power BI**.

3.1. ¿Qué es PARCE?

El **P**ortal **A**nalítico **R**ápido, **C**onversacional y **E**fectivo (**PARCE**) comprende un conjunto de herramientas en función de experiencias conversacionales para usuarios y empresas, es decir, **PARCE** brinda una asistencia conversacional a los usuarios y además brinda un ambiente analítico de la información conversacional a la empresa.



En detalle, PARCE está articulado por 3 pilares:

(1) **Asistente conversacional**: Es el servicio conversacional de PARCE, un servicio cognitivo entrenado para responder a las preguntas del usuario a través de un chatbot.

(2) **Almacenamiento de información**: PARCE tiene su propio lago de datos, en este repositorio se guarda la información de las métricas asociadas a las interacciones que hace el usuario con el asistente conversacional.

(3) **Ambiente analítico**: El ambiente analítico de PARCE consta de un espacio que usa los datos almacenados para desarrollar tableros de visualización y modelos analíticos para que la empresa pueda usar los datos que genera el asistente conversacional con el usuario.

Este proyecto integra habilidades de inteligencia artificial, almacenamiento de datos y desarrollo analítico, encaminadas a los siguientes objetivos:

- **Experiencia de usuario**: A través del desarrollo de PARCE explorar habilidades de IA para definir experiencias cognitivas dirigidas a la atención del usuario.
- **Agilidad analítica**: Diseñar y desplegar un ambiente de gestión de datos y desarrollo analítico para que la empresa tenga la facilidad de explorar y describir la información conversacional de los usuarios.
- **AI no tercerizada**: Explorar, investigar y documentar los servicios de tecnológicos de nube para establecer modelos de IA *inhouse*.

3.2. Preliminares de PARCE

En esta sección veremos conceptos claves y pasos previos de **PARCE** en el Data Sandbox.

En el Fondo Nacional del Ahorro cada día se atienden **miles** de usurios por medio de **canales presenciales y digitales**. En los digitales, los chatbots y servicios de mensajería captan información de los usuarios como: preguntas, dudas, quejas, necesidades especificas y tópicos de interés.

Luego, esta **captación** natural de información convierte a los canales digitales (correos electronicos, aplicativos y asistentes conversacionales) en **fuente principal de datos** de modelos descriptivos para generación de conocimiento y toma de decisiones con el propósito de atender las necesidades de los usuarios, además de ser insumo para desarrollar aplicaciones capaces de interactuar mejor con la población a través de canales de atención, esto por medio de entrenamiento de asistentes conversacionales **más eficientes**.

Tratamiento de los datos

Los **datos** con los que trabajamos en el proyecto PARCE son mensajes anonimizados de usuarios enviados a los canales de mensajería del **FNA**:

- Chatbot de preguntas frecuentes
- Whatsapp
- Correos electrónicos

Los mensajes recolectados tienen la siguiente forma:

	Mensaje de usuario
1	¿Cómo puedo obtener mi clave?
2	Necesito consultar el saldo de mis cesantias
3	¿Dónde encuentro los formularios para descargar?
4	¿Cómo hago para recuperar mi usuario y contraseña?
5	Necesito información del estado de un trámite
6	¿Puedo afiliarme por internet?

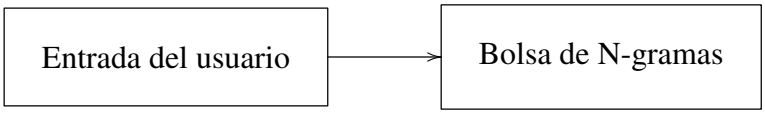
Por lo cual, como la *materia prima* son datos de texto donde no hay una estructura o segmentación definida de la información, es necesario usar modelos y herramientas especificas para el **análisis de texto**.

El **análisis de texto** es una técnica para extraer información valiosa del lenguaje humano de una manera inteligente y eficiente. [12] En otras palabras, es el proceso de deducir significado de textos y comunicaciones escritas, **con el fin de que estos datos semiestructurados puedan ser entendidos y analizados**.

En la siguiente tabla se presenta en brevedad el tratamiento que se ejecutó para la limpieza de datos de **PARCE**, con herramientas de análisis de texto.

Técnica	Descripción	Muestra
Captación	Texto original del usuario con el cual interactuó en el canal	¿Dónde encuentro los formularios para descargar?
Limpieza	Eliminar caracteres extraños, signos de puntuación y acentos	Donde encuentro los formularios para descargar
Normalización	Dejar todo el texto en minúsculas	donde encuentro los formularios para descargar
Stopwords	Eliminar palabras que dan poca significancia al texto	encuentro formularios descargar
Lematización	Transformar las palabras a su raíz gramatical	encuentr formular descarg
N-gramas	Asociar N-gramas a cada entrada. Los N-gramas son subsecuencias consecutivas en el texto	(encuentr formular) (formular descarg)

Del tratamiento anterior podemos definir la siguiente **asociación**:



Así, cada **entrada** que captamos del usuario y pasó por el tratamiento de análisis de texto, tendrá una **única** bolsa de N-gramas asociada. En otras palabras, al conjunto de datos (mensajes) que logran superar el tratamiento de datos descrito arriba, tendrá una nueva **caracterización** o **estructuración**, sus bolsas de N-gramas respectivas. Ahora, con una estructura definida (N-gramas) es más sencillo manipular los datos en **PARCE**.

Estructura de los datos

- ▷ **Hito**: Los datos de texto, inicialmente, son semiestructurados. Ahora, las bolsas de N-gramas permiten definir una estructura en los datos para identificar patrones y operar de forma más eficiente la información.

Con una estructura ya definida en nuestros datos, podemos identificar patrones y segmentaciones en los mismos. Para el caso de **PARCE**, el aprovechamiento fue lograr una **agrupación por tópicos**.

La agrupación por tópicos consiste en reunir todos los mensajes en grupos definidos de temas o tópicos según su afinidad por medio de las bolsas de N-gramas.

La idea es la siguiente:

(1) Definir un Diccionario de tópicos o temas de **mayor** interés para los usuarios.

- ▷ **Nota**: El diccionario de PARCE consta de **62 tópicos**, cada uno tiene su bolsa de N-gramas respectiva.

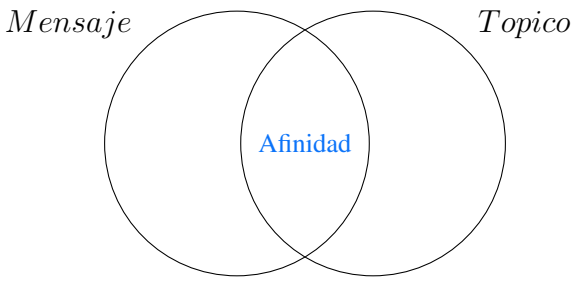
Tópico	Ejemplo PARCE
Tema 1	Afiliaciones
Tema 2	Consultar saldo
Tema 3	Certificaciones
Tema 4	Descargar factura

(2) Asociar una **bolsa de N-gramas** a cada tópico del diccionario.

Tópico	Ejemplo PARCE	Bolsa de N-gramas
Tema 1	Afiliaciones	(quier afilia), (necesit afilia)
Tema 2	Consultar saldo	(quier conoc sald), (necesit sald)
Tema 3	Certificaciones	(quier certific), (dond descarg certific)
Tema 4	Descargar factura	(dond descarg factur), (necesit factur)

(3) **Intersección de bolsas de N-gramas:**
El siguiente paso es muy importante y constah del **cruce** de bolsas de N-gramas de cada mensaje con cada tópico o tema del diccionario.

La idea inmersa en esto es **interceptar** cada **Bolsa de N-gramas** correspondiente al **mensaje** del usuario con cada **Bolsa de N-gramas** correspondiente al **tópico** del diccionario. Luego, la Afinidad se define como un nivel proporcional a la cardinalidad de la intersección de las dos bolsas, así, si la intercepción es de un solo N-grama podemos pensar en que la afinidad del mensaje y tópico es **baja**, por el contrario, si la intercepción es de varios N-gramas, concluimos una afinidad relevante entre el mensaje y el tópico.



- ▷ **Nota**: En el caso de PARCE tenemos:
  - ○ **240.000** mensajes de usuario, cada uno con una bolsa de N-gramas asociada.
  - ○ Un diccionario de **62** tópicos o temas, cada uno con una bolsa de N-gramas asociada.

  Por tanto, pensando en la intersección de bolsas de N-gramas para PARCE, deberíamos hacer un total de **14'880.000** intersecciones.

(4) **Umbral de afinidad:**

La afinidad entre las bolsas de N-gramas de mensajes y tópicos basicamente es un indicativo de una posible relación entre el mensaje de usuario y

un tópico del diccionario, sin embargo, un mensaje puede cruzar con varios tópicos, es decir, tener una afinidad positiva con varios tópicos del diccionario.

Como podemos encontrar múltiples afinidades entre un mensaje y los tópicos del diccionario, la mejor estrategia es definir un **umbral de afinidad** entre las bolsas de N-gramas, este umbral nos permite seleccionar las mejores intersecciones y así, las afinidades más cercanas a la realidad.

- ▷ **Nota**: En el caso de PARCE fijamos un umbral de 3 N-gramas, es decir, cuando la intersección supera este umbral, consideramos la afinidad. En caso contrario, no es una afinidad significativa para el modelo.

  La intersección de bolsas, naturalmente, puede ser nula, dado que el mensaje analizado es muy corto, o las bolsas de N-gramas de los tópicos no son lo suficientemente grandes.

(5) **Asignación del tópico:**

Finalmente, la mejor afinidad (el mensaje que mayor número de intersecciones haga con un tópico) es la que usamos para asignar un tópico al mensaje del usuario.

El proceso anteriormente descrito es la tecnica usada en PARCE para **agrupar** los miles de mensajes de usuarios en tópicos especificos del negocio.

Datos procesados

Hasta aquí ya tenemos un conjunto de datos procesados disponibles para el desarrollo de PARCE, en detalle, tenemos bases con miles de mensajes de usuarios con un tópico o tema asociado:

Mensaje	**Tópico**	Ejemplo PARCE
¿Puedo afiliarme por internet?	Tema 1	Afiliaciones
Necesito consultar el saldo de mis cesantias	Tema 2	Consultar saldo

3.3. Arquitectura

En esta sección compartimos los pasos para el despliegue de la arquitectura del proyecto PARCE del FNA en el Data Sandbox de MinTIC.

En cualquier proyecto de construcción tradicional, los detalles de infraestructura que se definen son muy importantes, como: ¿Cuántas bases debe tener un edificio para que soporte el peso de la construcción? o ¿De qué forma será la distribución de los pilares?, etc. Lo anterior no es más que la arquitectura de la construcción, siendo esta **la técnica para describir y diseñar** el proyecto.

Por otro lado tenemos, la **arquitectura de software**: que es el conjunto de entidades, propiedades, y las relaciones entre ellas que permiten crear y definir la estructura fundamental de un proyecto.

Cuando en la industria se habla de arquitectura, se refiere a una noción vagamente definida de los aspectos más importantes del diseño interno de un sistema de software. Una buena arquitectura es importante; de lo contrario, será más lento y costoso agregar nuevas capacidades en el futuro. (Martin Fowler, 2019).

Arquitectura de PARCE

En el caso de PARCE, para definir la arquitectura del proyecto, se tuvo en cuenta el siguiente propósito:

- ▷ Brindar un canal de comunicación de relacionamiento con el consumidor financiero que permita resolver las dudas acerca de los productos y servicios que presta la entidad, integrado a servicios de almacenamiento y analítica de datos.

Así que, en la arquitectura se deben identificar los componentes principales para **desarrollar** y poner en **producción** los siguientes 3 pilares que articulan a **PARCE**:

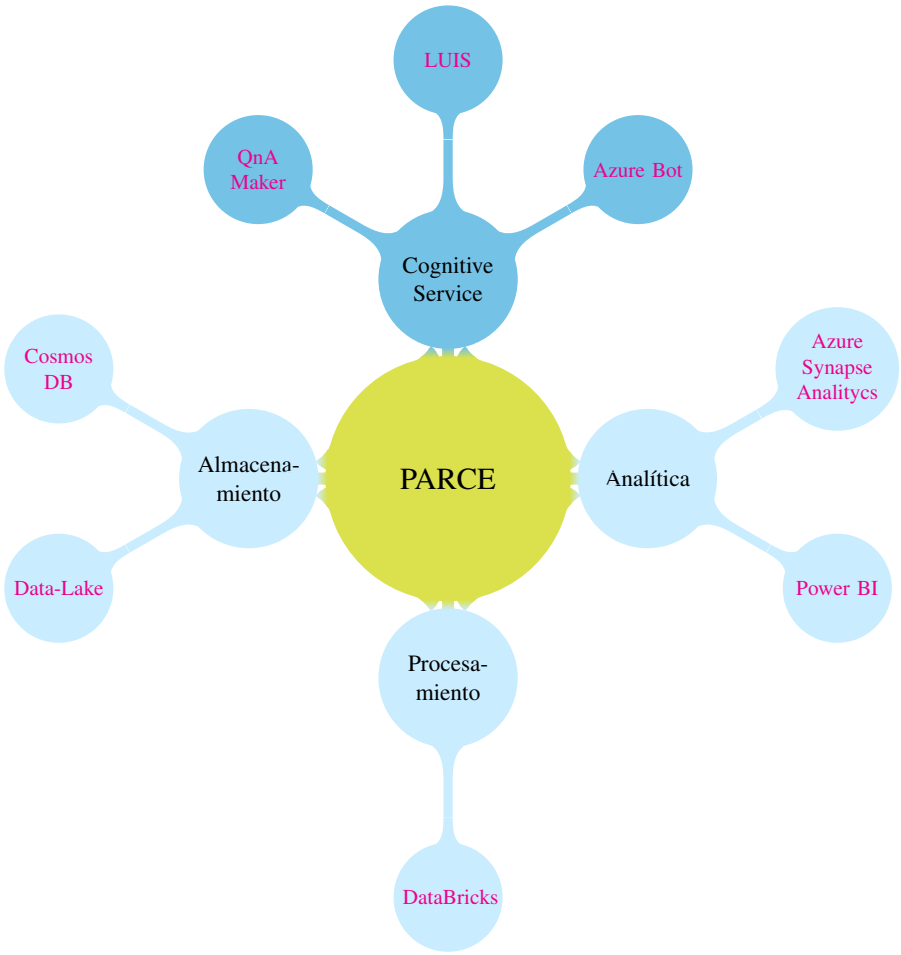
(1) Asistencia conversacional al usuario.

(2) Almacenamiento de datos conversacionales.

(3) Ambiente analítico de los datos conversacionales.

Para tal fin, en el marco del **Data Sandbox** aplicamos un proceso de **exploración de servicios**, para conocer y definir qué servicios de Azure formarían parte de la arquitectura de **PARCE**.

A continuación, presentamos la arquitectura de PARCE ideada con referencia en los servicios disponibles de Azure (*los nodos exteriores*).



En teoría, la arquitectura de PARCE está definida así:

- **Cognitive service**: Los servicios cognitivos de Azure permiten plasmar habilidades a proyectos para procesar información a partir de la percepción. En PARCE es trascendental el uso de estos servicios, ya que uno de sus pilares es la interacción conversacional con humanos.

  (1) **Azure Bot**: Es un entorno de desarrollo completo para diseñar y crear inteligencia artificial conversacional de nivel empresarial.

  (2) **QnA Maker**: Es un servicio de procesamiento de lenguaje natural (PLN) que permite crear una conversación natural con base en los datos.

(3) **LUIS**: Es un servicio conversacional de IA que aplica aprendizaje automático a una conversación o un texto de lenguaje natural de un usuario para predecir el significado global y extraer información de la misma.

- **Procesamiento**: Como sabemos, la generación de datos insumo para PARCE implica más de **14 millones** de intersecciones de bolsas de N-gramas, lo cual exige un nivel de procesamiento de maquina increible, para tal fin usamos **Data Bricks**.

  (1) **DataBricks**: Es un espacio de trabajo para el procesamiento de datos a gran escala.

- **Almacenamiento**: PARCE necesitará todo un entorno de almacenamiento de la información conversacional que capta en cada interacción de los clientes con el chatbot.

  (1) **Data Lake**: Es un repositorio centralizado que contiene grandes volúmenes de datos sin procesar.

  (2) **Cosmos DB**: Entorno de base de datos que permite distribuir datos de forma global.

- **Analítica**: En tiempo real será necesario mapear, operar y hacer uso de los datos captados en PARCE.

  (1) **Azure Synapse Analytics**: Es un servicio de análisis que acelera el tiempo necesario para obtener información de los sistemas de almacenamientos de datos.

  (2) **Power BI**: Entorno de creación de tableros de visualización.

Esta arquitectura no es más que una articulación de servicios de **Azure** que hacen posible la planeación, desarrollo y ejecución de PARCE.

En las secciones a continuación veremos como cada uno de los servicios que enlistamos antes hacen posible el funcionamiento de PARCE.

3.4. Data Lake

En esta sección presentamos el servicio **Data Lake** como herramienta de almacenamiento y storage de los datos usados para el desarrollo de **PARCE**.

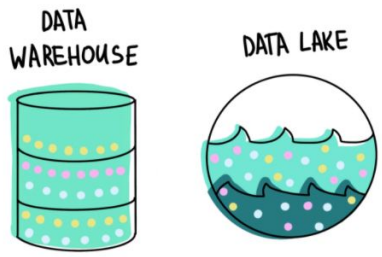


Figura 2: Data Warehouse y Data Lake

El almacenamiento en la nube es un servicio que permite almacenar datos transfiriéndolos a través de Internet o de otra red a un sistema de almacenamiento externo que mantiene un tercero. [2]

En los multiples servicios que existen en *la nube* para almacenamiento, existen conceptos como: **Data Lake** y **Data Warehouse**.

Data Lake y **Data Warehouse** se utilizan para el almacenamiento de grandes volúmenes de datos, sin embargo, a pesar de que ambos son repositorios de datos, conceptualmente no son lo mismo:

- Un Data Lake es un repositorio centralizado que contiene grandes volúmenes de datos sin procesar, que aún no tienen un propósito fijo.
- Un Data Warehouse es un repositorio de datos porcesados que tiene una estructuración y propósito de uso definido.

Algunas diferencias clave son [4]:

(1) Data Lake es un repositorio de almacenamiento para: datos estructurados, semiestructurados y no estructurados. Por otro lado, Data Warehouse es una estructura tecnológica disponible para el uso estratégico de datos.

(2) Data Lake define el esquema después de que se almacenan los datos, mientras que Data Warehouse define el esquema antes de que se almacenen los datos.

(3) Data Lake usa el proceso **ELT** mientras que Data Warehouse usa el proceso **ETL**.

Hasta aquí hemos visto:

- ▷ **Sección 3.2**: Conceptos preliminares con los cuales generamos los datos para el desarrollo del proyecto.
- ▷ **Sección 3.3**: Una visión rápida de cómo funciona la arquitectura diseñada para **PARCE**.

Ahora, veremos como usar el servicio **Data Lake** de Azure para almacenar las bases de datos insumo de PARCE, y así, poder usar estos datos en cualquier herramienta de Azure que lo disponga.

Data Lake en PARCE

Ya conocemos la forma de los datos insumo para el desarrollo de PARCE, básicamente, cadenas de texto (mensajes) con bolsas de N-gramas asociadas (caracterización del mensaje). Sin embargo, a pesar de definir una estructura a nuestro conjunto de datos base, su propósito en la *nube* de Azure en principio no era plenamente definido, por lo cual se optó por almacenarlos en un reposiorio de datos sin procesar, y sin próposito fijo de los mismos. Es decir, PARCE usa **Data Lake**.

Así, en el **grupo de recursos** de PARCE se desplegó el servicio: **Data Lake Storage**.

Este paso del proyecto no es más que la migración a la nube. Es la parte en la que tomamos nuestro material del ambiente local y lo cargamos a la nube, para poder usar nuestros datos en todos los servicios de Azure que los requieran.

Luego, tenemos los siguientes insumos para este proceso de cargue:

- ▷ **Base principal**: Base de mensajes de usuarios con bolsa de N-gramas asociado.
  - **Dimensión**: 240.000 mensajes.
  - **Tamaño**: 20 Mb.
- ▷ **Diccionario de tópicos**: Base de tópicos o temas referentes a las necesidades de los usuarios y preguntas frecuentes del negocio, cada uno con una bolsa de N-gramas asociada.

- **Dimensión**: 62 tópicos/temas.
- **Tamaño**: 13 Kb.

A continuación, veremos el proceso de *cargue* de los datos de insumo de PARCE a un **contenedor** del Data Lake Storage.

Crear un contenedor en Data Lake:

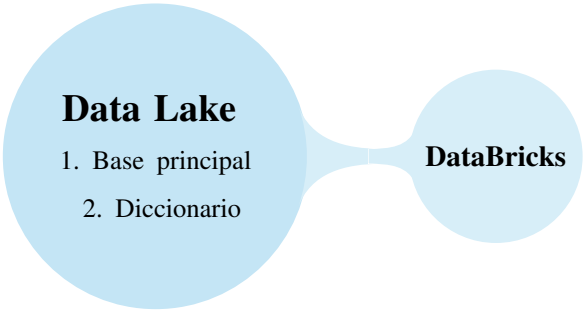
Un contenedor en Data Lake contiene directorios y archivos, en la etapa de exploración lo concluimos como la forma directa de cargar archivos en **'cajas'** con referencias básicas para su uso. Eje: *'Archivos iniciales', 'Resultados', etc...*

En Data Lake Storage existe una sección llamada Data storage, seleccionamos **Containers**, allí creamos un contenedor llamado **bases**, y procedemos a cargar nuestra **Base principal**: base de entradas de usuario con N-gramas y el **Diccionario de tópicos**.

En este punto, ya tenemos las bases de datos insumo para el desarrollo del proyecto **PARCE**.

El primer propósito que le daremos a los datos es el uso de los servicios de procesamiento de Azure.

- ▷ **Hito:** Bases de datos cargadas en un repositorio cloud, para uso en dicho ambiente.
- ▷ **Objetivo:** Usar el servicio de procesamiento DataBricks para ejecutar el proceso de intersección de bolsas de N-gramas, donde consuma las bases almacenadas en Data Lake.



3.5. DataBricks

En esta sección presentamos el uso de **DataBricks** para el *procesamiento* de las bases de datos, para creación de insumos de entrenamiento de los servicios cognitvos de **PARCE**.

▷ En la sección anterior realizamos el cargue de los insumos de datos a un contenedor del servicio **Data Lake Storage**.

En los *preliminares* del proyecto hicimos un cálculo abrumador para corresponder un tópico a los **240.000 mensajes**. Al respecto, consideremos lo siguiente:

(1) Cada bolsa de N-gramas (tanto del mensaje, como del tópico) tiene un número variante de N-gramas. Una bolsa de N-gramas no es más que una lista de N-gramas.

Por lo cual, la intersección de bolsas básicamente es una intersección de listas.

(2) Son **240.000** bolsas de los mensajes de usuarios, y **62** bolsas del diccionario de tópicos.

En total, son **14'880.000** intersecciones posibles.

(3) Previo al Data Sandbox, desarrollamos códigos en *python* que realizan lo sig:

i. Un bucle que hace TODAS las intersecciones de bolsas de N-gramas.

ii. Con base en un umbral de afinidad fijo, se asigna un tópico o tema a cada mensaje.

Sin embargo, el bucle hace una intersección cada **15** segundos (considere que las bolsas de N-gramas podrían crecer y así aumentar el tiempo de intersección).

(4) Conforme al punto anterior, en nuestro desarrollo local hacer TODAS las intersecciones posibles consumiría más de **223'200.000** segundos, alrededor de **62.000** horas.

Por lo cual, concluimos la imposibilidad de hacer la asignación de tópicos al conjunto de mensajes en el ambiente local. Así, convergemos al primer aprovechamiento del Data Sandbox: migrar los desarrollos locales a un ambiente de procesamiento en nube para acelerar tareas y obtener resultados.

DataBricks es una herramienta de ingeniería de datos basada en la nube que se utiliza para procesar y transformar cantidades masivas de datos y de diferente tipos para así explorar los datos a través de modelos de aprendizaje automático. [1]

DataBricks en PARCE

- ▷ **Reto:** Usar DataBricks como ambiente de procesamiento en nube para realizar la intersección de bolsas de N-gramas en un tiempo considerablemente menor al calculado en el ambiente local.

Antes mencionamos el desarrollo de códigos en python que estaban diseñados para hacer la tarea de intersección de bolsas, la idea es llevar dichos códigos a DataBricks.

En un principio pensamos que simplemente sería un ejercicio de cargar y ejecutar en el entorno de trabajo de DataBricks, sin embargo, descubrimos que el procesamiento de cantidades masivas de datos en DataBricks solo es posible por el procesamiento distribuido de datos, en donde nos topamos con un nuevo concepto: **Spark**.

Spark es un framework open source para la computación en paralelo utilizando clusters. Se utiliza especialmente para acelerar la computación iterativa de grandes cantidades de datos o de modelos muy complejos. [5]

En teoría, Spark permite **fraccionar o paralelizar** las tareas, es decir, el procesamiento se divide en *n máquinas*, y de esta manera, cuando se procesa una gran cantidad de datos cada máquina se encargará de una n-ésima parte del trabajo, y al final se une la producción.

Así que, para **PARCE** tuvimos que **transcribir** los códigos desarrollados en el ambiente local, al lenguaje Spark. Como el desarrollo local está en python, resolvimos transcribir los códigos de python a PySpark.

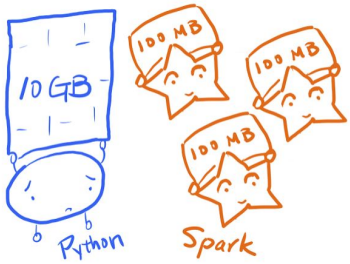


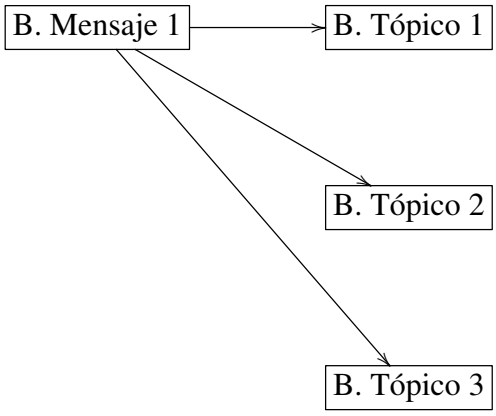
Figura 3: PySpark

Pyspark y funciones UDF

Pyspark es una biblioteca Spark escrita en Python para ejecutar la aplicación Python usando las capacidades de Spark, así se pueden ejecutar aplicaciones en paralelo en el clúster distribuido (múltiples nodos). En otras palabras, Pyspark es una API de Python para Spark. [13]

- ▷ **Reto:** Transcribir el código de intersección de bolsas de N-gramas desarrollado en Python, a Pyspark, para usar el procesamiento distribuido de DataBricks.

La transcripción de los códigos de etiquetado de mensajes fue sencilla dada la buena documentación en internet acerca de equivalencias entre lo que escribimos en Python con Pyspark. Sin embargo, hubo un cambio importante al script, este fue el uso del `for` para interceptar **dos a dos** las bolsas de N-gramas.



Es decir, con el `for` cada Bolsa de mensaje (*de los 240.000*) se intercepta con cada Bolsa de tópico (*de los 62*).

En **spark** podemos definir funciones UDF, para usar en el script y aprovechar el procesamiento distribuido.

Las funciones UDF (*User Defined Functions*) son las funciones de usuario, y son sistemas para definir nuevos métodos SQL que operan sobre las columnas de un DataFrame en spark. [11]. Así que, para ejecutar de forma distribuida las tareas de etiquetado en nuestro proyecto, se deben usar funciones UDF en el caso de tareas iterativas. Por lo cual, el `for` definido para hacer la instersecciones de Bolsas de N-gramas lo reemplazamos por la función `map` para definirla en una UDF.

- ▷ La función map() en Python aplica una función a cada uno de los elementos de una lista, que en lugar de aplicar una condición a un elemento de una lista o secuencia, aplica una función sobre todos los elementos.

Datos entrenados

Ya con todo el ambiente de ejecución listo y funcional en DataBricks, nos preparamos para importar las bases de insumo y hacer el etiquetado de los **240.000** mensajes de usuarios.

De la creación de los datos de entrenamiento, resaltamos los siguientes puntos:

(1) **Conexión DataBricks - Data Lake**:
Los datos que consume DataBricks son importados desde el contenedor creado en Data Lake, en donde almacenamos la base de mensajes y el diccionario de tópicos.

- ▷ En los códigos desarrollados en Pyspark de DataBricks se enseña cómo importar y exportar datos en DataBricks, desde y hacia Data Lake.
- ▷ Puede encontrarlos en el repositorio GitHub del proyecto PARCE.

(2) **Velocidad de procesamiento**:
Algo más de **62.000** horas eran necesarias para hacer TODAS las intersecciones posibles entre bolsas de N-gramas. Al transcribir el script a Pyspark para procesar la tareas de forma distribuida, solo se necesitaron **5** minutos para hacer la misma cantidad de intersecciones y crear los datos de entrenamiento para PARCE.

(3) **Afinidad nula**:
Como mencionamos en *preliminares* la intersección entre bolsas de N-gramas puede ser nula, es decir, hay bolsas de mensajes que no interceptan con ninguna bolsa de N-gramas de tópicos.

Por lo cual, hay mensajes que no lograron tener etiqueta.

(4) **Porcentaje de etiquetado**:
Recordemos que la base inicial para etiquetar tiene **240.000** mensajes. Despúes del procesamiento distribuido para las **14'880.000** intercepciones, fueron **124.000** mensajes que recibieron un tópico por el algoritmo de etiquetado.

- ▷ El porcentaje de etiquetado de nuestros datos fue de **52 %**, en otras palabras, con el proceso ejecutado en DataBricks logramos asignarle un tópico a 124.000 mensajes de usuarios.

(5) **Datos entrenados**:
Hasta aquí hemos logrado etiquetar mensajes con algún tópico del diccionario.

La asignación de un tópico o tema a cada mensaje de usuario nos permite manipular todo el volumen de mensajes de forma hábil y útil. Para PARCE el hecho de conocer el tópico asignado a un mensaje, permite también asignar una **respuesta** a cada mensaje.

Así que, conocer las correspondecias mensaje - tópico permite crear las correspondencias mensaje - respuesta.

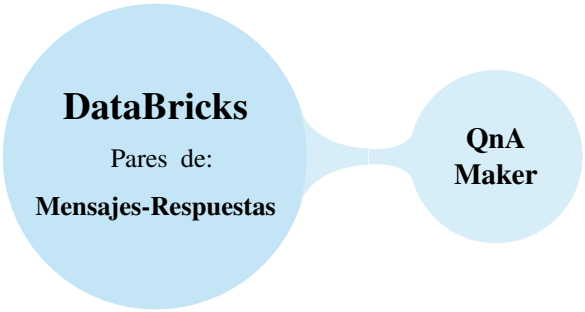
Mensaje	**Tópico**	**Respuesta**
¿Puedo afiliarme por internet?	Tema 1	Respuesta 1
Necesito consultar el saldo de mis cesantias	Tema 2	Respuesta 2

Simplemente, cada tópico tendrá una respuesta asociada, y cuando el mensaje tenga una afinidad con algún tópico, directamente tendrá afinidad con la respuesta asignada al tópico.

Finalmente, hemos producido datos de entranamiento para usarlos en las herramientas cognitivas de Azure.

▷ **Hito**: Como vimos, en **PARCE** era necesario hacer **14'880.000** de iteraciones en una tarea para la creación de datos de entrenamiento, dicha tarea en ensayos previos en maquinas locales tardaría cerca de 62.000 horas (aprox. 7 años).

Con los códigos de intersección de bolsas de N-gramas *transcritos* a **Pyspark** logramos aprovechar el procesamiento distribuido en **DataBricks** y la tarea que antes tardaría 7 años, finalmente tardó 5 MINUTOS en ejecutarse.



3.6. QnA Maker y LUIS

En esta sección presentamos el uso de los servicios cognitivos **QnA Maker** y **Language Understanding (LUIS)** para la creación de las habilidades conversacionales de **PARCE**.

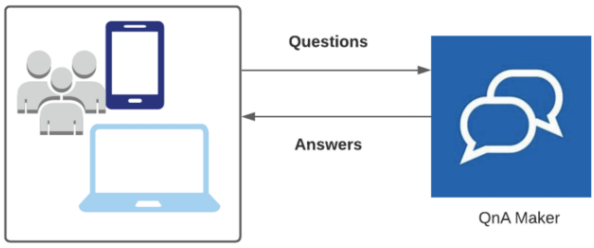


Figura 4: **QnA Maker** de Azure

Uno de los pilares principales de PARCE es la asistencia conversacional al usuario, es decir, la interacción de primera línea con los clientes que buscan comunicarse o interectuar con el canal de atención.

QnA Maker es un servicio de Procesamiento de Lenguaje Natural (**PLN**) que permite crear un ambiente de conversación natural con base en los datos. **QnA Maker encuentra** la respuesta más apropiada para una entrada de la base de conocimiento personalizada (Knowledge Base) de información. [7]

- ▷ QnA Maker **RESPONDE** con base en un repositorio de pares de pregunta-respuesta que se le asigne como base de conocimiento.

QnA Maker en PARCE

Naturalmente, PARCE debe brindar asistencia a las dudas y preguntas de usuarios, esto es posible por medio de los servicios cognitivos de Azure, puntualmente, con **QnA Maker**.

Recordemos que ya tenemos algo más de **124.000** pares de mensaje - respuesta, resultado de etiquetar la base de los mensajes con algún tópico del diccionario de temas y tópicos. Y como bien expresamos al final de la sección anterior, esta base de etiquetados son nuestros **datos de entrenamiento**.

El uso del servicio cognitivo **QnA** para PARCE lo describimos en los siguientes

hitos:

(1) **Crear una KB**
El paso inicial es crear una base de conocimiento personalizada segun el propósito que desea darle a QnA Maker.

La forma en la que QnA Maker importa los datos de entrenamiento es lo que conocemos como una base de conocimiento de pares de **preguntas y respuestas**.

- ▷ Importamos los **124.000** pares de mensaje-respuesta creados en la sección de procesamiento de DataBricks.
- ▷ Crear un KB en QnA Maker es muy sencillo, puede saber cómo siguiendo el siguiente enlace: https://www.qnamaker.ai/Create

(2) **Entrenar una KB**
Como cualquier servicio de inteligencia artificial es importante **entrenar** la herramienta conforme a la atención particular del negocio. En el caso de PARCE debiamos entrenar la KB para que respondiera coherentemente a múltiples entradas de usuarios con dudas referentes a productos y servicios del FNA.

(3) **Testear una KB**
Básicamente, despúes del entrenamiento se deben hacer pruebas para testear la calidad de la KB, la forma de hacerlo es insertar preguntas referentes a la KB y de diferentes formas para así reentrenar y extender la capacidad de entendimiento de la KB.

- ▷ La KB califica la elección de respuesta que enviará al usuario con un **score** de confianza, así podemos reentrenar pares de pregunta-respuesta para hacer una KB más precisa y coherente, con base en buenos scores de elección.

  Se pueden hacer múltiples entrenamientos y testeos a una KB.
- ▷ Luego de entrenar y testear una KB, el paso a seguir es **publicar** la KB en el servicio de QnA Maker (*este paso nos dará contraseñas para conectar la KB a otros servicios de Azure*).
- ▷ Justamente en Azure Bot podremos vincular la KB publicada en QnA Maker y hacer pruebas de preguntas y respuesta.

  Si lo nota, en este punto ya hemos creado un **ChatBot** de preguntas y respuestas con base en los datos entrenados para PARCE.

(4) **Retos**
Hasta este punto ya: creamos, entrenamos, testeamos y publicamos una KB para PARCE. Sin embargo, en el proceso nos enfrentamos con 2 **retos** para poder tener una buena KB:

- **Forma de la pregunta**:
  En QnA Maker es importante la forma en la que está escrita la pregunta en la KB. Generalmente si creamos una KB en español, la forma de la pregunta en la KB debe ser fiel a la estructura gramatical de las preguntas en español, es decir, con el correcto uso de los signos de interrogación y con el estilo correcto de una pregunta.

  Ahora, nuestro problema es el siguiente: Las entradas de usuario de nuestra KB son interacciones reales de usuarios del FNA, en donde, NO siempre se sigue fielmente la estructura gramatical de una pregunta.

- **Límites de QnA Maker**:
  Las KB de QnA Maker tienen límites con los que nos vimos enfrentados en el momento de la creación de la KB. Puntualmente, el número de formas alternativas de una pregunta puede ser en promedio de **300**.

  Ahora, nuestro problema es el siguiente: En la etapa de etiquetado logramos asignar algún tópico a **124.000** mensajes, luego, los tópicos quedaron asignados en promedio a **4000** mensajes, es decir, hay respuestas con **4000** alternativas de pregunta en nuestra KB.

Las soluciones que encontramos a estos retos fueron:

Primero, plantear una limpieza extra a las entradas de usuario que recibieron etiqueta, para mejorar su forma gramatical o dejar las que tuviesen una fiel forma gramatical de una pregunta.

Segundo, establecer un umbral de afinidad más estricto, para dejar a lo sumo las mejores **150** afinidades por tópico.

Por otro lado, el detalle revelador es que NO usamos los **124.000** pares de mensaje-respuesta que habiamos producido, usamos las mejores afinidades.

Finalmente, publicamos una KB para PARCE que conectamos a **Azure Bot**, pero la conexión importante es a Framework SDK de nuestro Bot (*detallamos esto en la siguiente sección*).

▷ **Hito**: Concluimos que **NO** hay una última versión o la mejor versión de una KB, siempre es posible reentrenar y mejorar la KB para el servicio de QnA Maker. De hecho, para PARCE creamos varías KB, y nos quedamos con la que parcialmente trabajaba más coherentemente.

LUIS en PARCE

▷ **Hito**: **PARCE** ya cuenta con un servicio cognitivo que responde a las preguntas de usuarios, aprendiendo a responder con base en la información captada en los canales digitales del FNA.

Por un lado tenemos a QnA como servicio de interacción cognitiva de PARCE, lo cual define el primer pilar del proyecto: **Asistencia conversacional**. QnA Maker tiene la tarea de **RESPONDER** a la interacciones de los usuarios de PARCE, con una base de conocimiento personalizada. Sin embargo, hay otro servicio cognitivo de nuestro interés, **Language Understanding** (LUIS).

LUIS es un servicio conversacional de inteligencia artificial que aplica inteligencia de aprendizaje automático personalizado a una conversación o un texto de lenguaje natural de un usuario para predecir el significado global y extraer información pertinente y detallada. [9]

▷ LUIS **PREDICE** la intención del usuario con base en la interacción.

Nuestro propósito ahora es implementar **LUIS** en PARCE. La idea es aprovechar la función de **predicción** de un texto de entrada de un usuario, además, será muy interesante que en PARCE trabajen a la par dos servicios cognitivos, y así **comparar** el funcionamiento de los mismos.

Recordemos que tenemos **62** temas o tópicos en el diccionario que usamos para etiquetar o asignar un tópico a nuestra base de mensajes de usuarios. Ahora, usaremos dicho diccionario en el entrenamiento de LUIS para que este servicio cognitivo pueda predecir de los mensajes si la intención está asociada con alguno de los **62** tópicos.

El uso del servicio cognitivo **LUIS** para PARCE lo describimos en los siguientes logros:

(1) **Crear una app LUIS**
Como con QnA Maker, la idea es crear una aplicación LUIS específica para PARCE.

▷ Como **LUIS** predice intenciones, las intenciones que usaremos en PARCE serán justamente los 62 tópicos de interés con los que etiquetamos los mensajes de usuario.

▷ Crear un app en LUIS es muy sencillo, puede documentarse sobre cómo hacerlo en el siguiente enlace: https://www.luis.ai/applications

(2) **Entrenar y testear**
De formar similar que en la KB de **QnA Maker**, la app LUIS es una inteligencia que debemos entrenar y testear.

▷ Entrenamos **LUIS** con muestras por cada intención de las **62** definidas para PARCE. Es decir, **LUIS** será capaz de predecir a cual de las **62** intenciones definidas corresponde la entrada del usuario.

▷ El testeo de la app es similar que con una KB, básicamente debemos ingresar alternativas de mensajes para poder medir la coherencia y precisión de **LUIS**.

▷ Después de entrenar y testear, **publicamos** la app para poder usarla en otros servicios de Azure.

(3) **Retos**
Hasta este punto ya: creamos, entrenamos, testeamos y publicamos una **app LUIS** para PARCE. Sin embargo, en el proceso nos enfrentamos con 1 reto adicional para poder obtener el mayor provecho de LUIS:

- **Manualidad en el entrenamiento:**
  El proceso para crear una intención y darle muestras de entrenamiento es manual, es decir, debemos escribir una a una las entradas de entrenamiento.

  Nuestro problema yace en lo siguiente: Al tener miles de muestras para enseñar a LUIS y no tener la posibilidad de cargar de forma masiva dichas muestras, el proceso manual es ineficiente.

La solución que propusimos fue la siguiente: Escribir manualmente entradas de entrenamiento con buena calidad estructural y gramatical, basándonos en algunas muestras de las entradas reales.

Como en el caso de QnA Maker, no hay una app LUIS definitiva, siempre se puede reentrenar y mejorar para aumentar su promedio de score de predicción.

Finalmente, **PARCE** dispone de 2 servicios cognitivos para fundamentar su pilar de asistencia conversacional.

3.7. Bot Framework y Cosmos DB

En esta sección presentamos el uso del **BotFramework** para el diseño interno del ChatBot de PARCE, y la conexión con una base de datos para el almacenamiento de la información del Bot de PARCE, esto usando **Cosmos DB**.

El **Bot Framework**, junto con **Azure Bot Service**, proporciona herramientas para compilar, probar, implementar y administrar bots inteligentes.

Bot Framework incluye un **SDK** modular y extensible para crear bots, así como herramientas, plantillas y servicios de inteligencia artificial relacionados. Así, se pueden crear bots que usan voz, comprenden el lenguaje natural, controlan preguntas y respuestas, etc. [8]

- **Hito:** El proyecto tiene un servicio cognitivo, **QnA Maker**, que responde a preguntas referentes a la base de conocimiento de PARCE, este servicio de respuestas está conectado a **Azure Bot Service**.

Así que, PARCE cuenta con un **ambiente conversacional** funcional, que responde a preguntas relacionadas a los tópicos o temas del diccionario usado para la creación de los datos de entrenamiento.

Sin embargo, en QnA Maker NO encontramos la posibilidad de **registrar** o **almacenar** el mensaje del usuario que usa el ChatBot de PARCE.

- El servicio de **Azure Bot** tampoco facilita una opción para registrar en una base de datos las interacciones con el ChatBot conectado a QnA Maker.

Justamente el segundo pilar de PARCE es el **almacenamiento de información** conversacional del usuario.

Bot Framework en PARCE

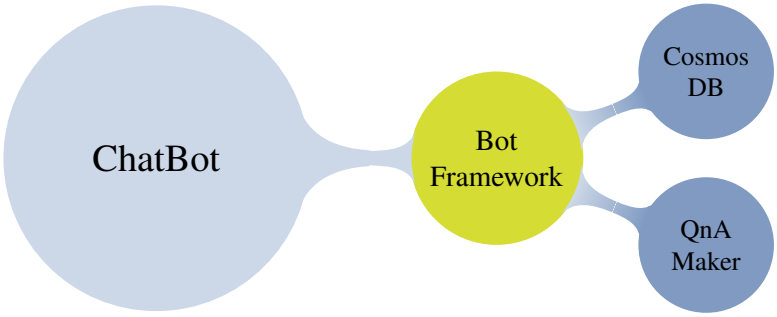
En PARCE usamos el SDK de **Bot Framework** con el siguiente propósito: Crear un bot que este conectado a una **KB** de **QnA Maker** para responder preguntas, y que además este conectado a un servicio de almacenamiento para guardar métricas de las interacciones de los usuarios con el ChatBot de PARCE.

Para la creación del Bot que **responde** y **almacena**, usamos:

- El repositorio en GitHub de código abierto con multiples plantillas y funcionalidades para el desarrollo de bots, en particular, nos interesa que el bot de PARCE responda con su **KB** de QnA Maker y almacene información conversacional.

- ▷ **Cosmos DB**, servicio que desplegamos y en donde creamos una base de datos para **PARCE**.

Representamos la idea en el siguiente dibujo:



El Bot que creamos **combina** dos habilidades en su estructura:

(1) **Responder**: Usamos una plantilla de bot que se conecta a una KB de **QnA Maker**, y tiene código *Python* escrito para responder con base en dicha KB.

(2) **Almacenar**: Usamos una plantilla de bot que se conecta a una base de datos de **Cosmos DB**, y tiene código *python* escrito para almacenar información conversacional en dicha base de datos.

En detalle, logramos almacenar las siguientes métricas de un mensaje de usuario: mensaje de usuario, respuesta del bot, fecha, hora e indicador si el bot pudo responder a la pregunta.

Básicamente, combinamos las dos habilidades en un solo código que permitiera el aprovechamiento de **almacenamiento** y **respuesta** en un solo ambiente.

En la ejecución de esta parte del proyecto, nos encontramos con 2 puntos relevantes:

(1) **Despliegue**:

Naturalmente, como creamos un bot en el ambiente local con el SDK de Bot Framewrok, el paso a seguir es desplegar el desarrollo local a la **nube** con un servicio para la publicación de dicho desarrollo, **Azure Bot Service**.

El despliegue **NO** fue sencillo ya que la documentación acerca de despliegue de proyectos a Azure Bot Service está escrita para desarrollos en Csharp, y nuestro desarrollo está en **python**.

(2) **Almacenamiento**:

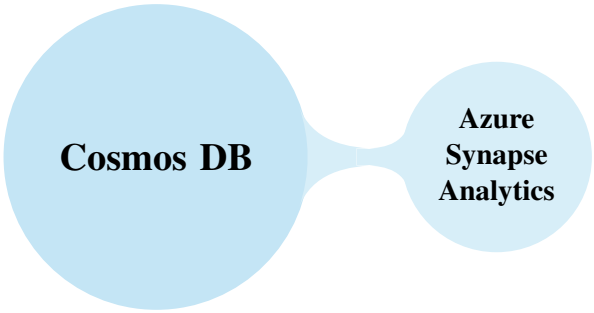
Las métricas conversacionales del bot de PARCE, por defecto Cosmos DB las almacena como archivos **.json** y de forma **individual**. Es decir, no las almacena como una tabla de filas - columnas.

Esto implica un reto para el desarrollo analítico prático sobre los datos.

En este punto, PARCE cuenta con **2** de sus pilares importantes:

Asistencia conversacional, logrado con los datos de entrenamiento creados usando el procesamiento distribuido en **DataBricks** para diseñar un asistente conversacional en QnA Maker que responde a preguntas de usuario relacionadas con una base de conocimiento personalizada. **Almacenamiento de información** conversacional, logrado con un bot diseñado y creado con el SDK Bot Framework de Microsoft, el cual se conecta con **QnA Maker** para articular la asistencia conversacional, y por otro lado, almacenar métricas especificas de cada interacción hecha en el ChatBot en una base de datos de **Cosmos DB**.

Finalmente, ya tenemos captación de datos de nuestro propio asistente conversacional, disponible para hacer analítica de datos en **PARCE**.



3.8. Azure Synapse Analytics

En esta sección presentamos el uso de **Azure Synapse Analytics** para el diseño del **ambiente analítico** de PARCE.

El inicio de este proyecto consistió en **describir** miles y miles de mensajes de usurios que interactuaron con algún canal no presencial de la Entidad, básicamente para conocer los temas de interés consultados por los usuarios del canal.

- ▷ **PARCE** no puede ser ajeno a una analítica de datos para la toma de decisiones.

Azure Synapse Analytics es un servicio de **análisis** ilimitado que reúne la integración y el almacenamiento de datos, así como el análisis de macrodatos. Azure Synapse Analytics ofrece una experiencia unificada para **ingerir**, **explorar**, **preparar**, **transformar**, **administrar** y **servir datos** con el fin de satisfacer las necesidades inmediatas de inteligencia del negocio y aprendizaje automático. [6]

Synapse Analytics en PARCE

El almacenamiento de datos de PARCE hace uso de **Cosmos DB**, que como mencionamos en la sección anterior, almacena interacciones de forma **individual** y en formato **.json**. El reto es construir una tabla que agrupe las interacciones con sus métricas, captadas en la base de datos de Cosmos DB.

Presentamos una muestra de una interacción del ChatBot de **PARCE** almacenada en Cosmos DB.


```
{
    "id": "aep6iwTR3s",
    "realId": "aep6iwTR3s",
    "document": {
        "py/object": "bots.qna_bot.UtteranceLog",
        "turn_question": "gracias",
        "turn_answer": "Con gusto.",
        "turn_date": "20-09-21",
        "turn_time": "20:21:32",
        "turn_understood": "S"
    }
}
```


El proceso de transformación de los registros almacenados se resume en estos pasos:

(1) **Espacio de trabajo**:
Después del despliegue de Azure Synapse Analytics creamos **notebooks** para la ingesta y transformación de los resgistros captados por el ChatBot de PARCE.

- ▷ Al igual que en DataBricks, en Azure Synapse desarrollamos en **Pyspark** para aprovechar el procesamiento distribuido.

(2) **Conexión base de datos**:
Los resgistros captados por el ChatBot están almacenados en una base de datos de **Cosmos DB**, para ejecutar el proceso, debemos importar dichos datos desde Azure Synapse Analytics. Hay varias formas de hacer la conexión, ya sea desde Cosmos DB a Azure Synapse Analytics, o viceversa.

(3) **Transformación de registros**:
Con Pyspark aprendimos buenas formas de procesar en paralelo o de forma distribuida, en especial, usar funciones que facilitaran este tipo de procesamiento.

- ▷ **Hito:** Con la función `map` transformamos cada resgistro **.json** en una fila de una tabla. De esta forma, aprovechamos el procesamiento distribuido en Azure Synapse Analytics.

(4) **Transformación de datos**:
En el desarrollo del bot en SDK de Bot Framework detallamos las métricas que se exportarían a la base de datos: mensaje de usuario, respuesta del bot, fecha, hora e indicador si el bot pudo responder a la pregunta. Sin embargo, para PARCE es de gran utilidad **relacionar** a cada intereacción almacenada la etiqueta o servicio del diccionario, para así tener en cuenta la **coherencia** de **QnA Maker**.

- ▷ El código para lograr dicho relacionamiento fue escrito en Pyspark, básicamente es hacer un 'merge' especial con un diccionario modificado para Synapse.

(5) **Automatización**:
En este punto, el notebook hace las siguientes tareas:

(i) Importa los registros captados por el ChatBot de una base de datos de Cosmos DB.

(ii) Transforma los registros de .json a una tabla.

(iii) Relaciona los registros almacenados con una nueva métrica: tópico del diccionario.

(iv) Guarda la tabla.

(v) Exporta la tabla a un contenedor de Data Lake.

Sin embargo, la ejecución del notebook en un principio se hacia manualmente, así que, para que PARCE no tenga este proceso manual creamos un **Pipeline - Triggers**.

Una canalización ejecutada en Azure Synapse define una **instancia de ejecución** de una canalización. Por ejemplo, supongamos que tiene una canalización que se ejecuta a las 8:00 a.m todos los domingos. [10] Así que, todos los domingos a las 8:00 a.m el notebook se ejecutará automaticamente.

▷ Para **PARCE** definimos un **Pipeline - Trigger** todos los domingos a las 10 pm. Es decir, el notebook descrito arriba se ejecuta automaticamente cada semana.

Aquí, PARCE cuenta con una tabla que describe todas la interacciones del ChatBot con métricas conversacionales.

Enriquecimiento de los datos

Actualmente, el almacenamiento de información conversacional de PARCE cuenta con métricas básicas como:

(1) Entrada del usuario.

(2) Respuesta del bot.

(3) Fecha y hora.

(4) Indicador de entendimiento del bot.

(5) Etiqueta o tópico relacionado al diccionario.

Ahora bien, el proceso de enriquecer los datos consiste en asignar nuevas métricas a la tabla. El enriquecimiento que hicimos fue: Asignar la predicción de la aplicación **LUIS** que creamos en unas secciones atrás. Para tal fin, usamos el **SDK** de LUIS, para usar la aplicación como una API y consultar la predicción de cada **mensaje** de nuestra tabla.

▷ **Hito**: PARCE tiene dos servicios cognitivos funcionando en su ambiente analítico. Por un lado **QnA Maker** que responde y asocia la etiqueta que elije para responder, y por otro lado, **LUIS** que toma el mensaje y da una predicción de la intención real del usuario.

- ▷ **Hito**: En PARCE podemos comparar 2 servicios cognitivos que trabajan en el mismo ambiente.

Finalmente, PARCE cuenta con **todos** sus pilares:

Asistencia conversacional, logrado con los datos de entrenamiento creados usando el procesamiento distribuido en **DataBricks** para diseñar un asistente conversacional en QnA Maker que responde a preguntas de usuario relacionadas con una base de conocimiento personalizada. **Almacenamiento de información** conversacional, logrado con un bot diseñado y creado con el SDK Bot Framework de Microsoft, el cual se conecta con **QnA Maker** para articular la asistencia conversacional, y por otro lado, almacenar métricas especificas de cada interacción hecha en el ChatBot en una base de datos de **Cosmos DB**. **Ambiente analítico**, logrado con **Azure Synapse Analytics**, el cual se conecta con **Cosmos DB** para consumir la información captada por el ChatBot, y disponer los datos para el desarrollo de modelos analíticos y proyectos de visualización.

4. Resultados

En esta sección presentamos los resultados del desarrollo de **PARCE** en el Data Sandbox.

Abordamos **4** hitos importantes como resultados: El procesamiento de datos, la creación de asistencia conversaciona con **IA**, el almacenamiento de información y la ingesta de datos en un ambiente analítico.

- ▷ **Velocidad de procesamiento**
  En menos de **5 minutos** logramos procesar una tarea de más **14 millones** de iteraciones, que tardaba más de **62.000** horas en la maquina local. Esto fue posible usando el procesamiento distribuido en DataBricks.

  Servicios usados: DataBricks y Data Lake.

- ▷ **IA para la asistencia conversacional**
  Con nuestros propios datos de entrenamiento creamos 2 servicios cognitivos que se comportan acorde a los datos con los que se entrenaron. **QnA Maker** que RESPONDE a la interacciones de los usuarios de PARCE. Y **LUIS** que PREDICE la intención del usuario con base en su mensaje.

  Servicios usados: QnA Maker, LUIS, Bot Framework y Azure Bot Service.

- ▷ **Almacenamiento de información**
  Con el SDK de Bot Framework diseñamos un bot que responde con base en el servicio entrenado en **QnA Maker**, y además, almacena la información captada por el bot: la interacción y métricas de la misma, en una base de datos desplegada en **Cosmos DB**.

  Servicios usados: Cosmos DB y Bot Framework.

- ▷ **Ambiente analítico de datos**
  Con una captación de información conversacional ya estructurada y funcional, creamos un ambiente de ingesta de dicha información. Con **Azure Synapse Analytics** conformamos un ambiente en donde: transformamos y enriquecemos las métricas de la información conversacional del ChatBot, para así, disponer los datos conversacionales de PARCE para modelos analíticos y en particular, para tableros de visualización en **Power BI**.

  Servicios usados: Azure Synapse Analytics, Cosmos DB, Data Lake y LUIS SDK.

5. Conclusiones

A continuación compartimos conclusiones del desarrollo de **PARCE** en el **Data Sandbox** del FNA.

(1) El Data Sandbox de MinTIC es un vehículo de innovación que permite a las empresas públicas acceder a servicios de nube, capacitaciones y acompañamiento por expertos en Bigdata para responder preguntas de negocio que requieran de estas tecnologías.

(2) El aprendizaje continuo y la oportunidad de ser certificado por los fabricantes, en este caso Microsoft, son pilares importantes en la consolidación de equipos expertos en materia de Bigdata.

(3) La participación del Equipo de analítica del FNA en el Data Sandbox le permitió procesar grandes volúmenes de información, explorar, servicios, capacitarse, experimentar y acceder a acompañamiento para la creación de un chatbot.

(4) Las ventajas de procesamiento de maquina en *cloud* son significativamente superiores a la capaciadades locales, en **PARCE** experimentamos y concluimos la oportunidad inmensa de procesar datos en el ambiente *cloud*. Simplemente con codificación en **Pyspark** para el procesamiento distribuido en DataBricks logramos convertir una tarea de **7 años** a una tarea de **5 minutos**.

(5) Los servicios cognitivos que dispone Azure son funcionales y practicos (*con una base de conocimiento creamos un chatbot de preguntas y respuestas con QnA Maker*), sin embargo identificamos una mejora de integración con servicios de almacenamiento de datos conversacionales, e integración entre los mismos servicios cognitivos *QnA Maker y LUIS*.

En **PARCE** integramos QnA Maker con un servicio de almacenamiento usando el sdk de Bot Framework, e integramos a LUIS y QnA Maker con el sdk de LUIS en Azure Synapse.

(6) Con el Data Sandbox de MinTIC logramos desarrollar una aplicación de atención conversacional y analítica conversacional integrando servicios de IA, almacenamiento, procesamiento, y analítica en Azure; haciendo del uso de servicios cloud para la creación de soluciones de Bigdata una herramienta alcanzable y disponible para cualquier entidad pública.

Referencias

[1] Agenciab12. *'Qué es Azure Databricks'*. Nov. de 2021. URL: https://agenciab12.com/noticia/que-es-azure-databricks.

[2] Microsoft Azure. *'Qué es el almacenamiento en la nube y cómo se utiliza'*. Nov. de 2021. URL: https://azure.microsoft.com/es-es/overview/what-is-cloud-storage/.

[3] GOV.CO Datos Abiertos. *'Historia proyectos Datos Abiertos, Datos Abiertos Colombia'*. Nov. de 2021. URL: https://www.datos.gov.co/stories/s/vpr2-fnas.

[4] Guru99. *'Data Lake vs Data Warehouse: What's the Difference?'* Nov. de 2021. URL: https://www.guru99.com/data-lake-vs-data-warehouse.html.

[5] Medium. *'Pyspark manejo de funciones'*. Nov. de 2021. URL: https://medium.com/datos-y-ciencia/pyspark-manejo-de-funciones-507284716018.

[6] Microsoft. *'Azure Synapse Analytics'*. Nov. de 2021. URL: https://azure.microsoft.com/es-es/services/synapse-analytics/.

[7] Microsoft. *'Documentación de QnA Maker'*. Nov. de 2021. URL: https://docs.microsoft.com/es-es/azure/cognitive-services/QnAMaker/.

[8] Microsoft. *'Documentación del SDK de Bot Framework - Bot Service'*. Nov. de 2021. URL: https://docs.microsoft.com/es-es/azure/bot-service/index-bf-sdk?view=azure-bot-service-4.0.

[9] Microsoft. *'Introducción a Language Understanding (LUIS)'*. Nov. de 2021. URL: https://docs.microsoft.com/es-es/azure/cognitive-services/luis/what-is-luis.

[10] Microsoft. *'Pipeline execution and triggers in Azure Data Factory or Azure Synapse Analytics'*. Nov. de 2021. URL: https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers.

[11] OpenWebinars. *'UDFS en Spark SQL'*. Nov. de 2021. URL: https://openwebinars.net/blog/udfs-en-spark-sql/.

[12] QuestionPro. *'Text analysis'*. Nov. de 2021. URL: https://www.questionpro.com/tour/text-analysis.html.

[13] Spark. *'PySpark Tutorial For Beginners'*. Nov. de 2021. URL: https://sparkbyexamples.com/pyspark-tutorial/.