



Tecnológico de Monterrey

**Instituto Tecnológico y de Estudios Superiores de Monterrey,
Campus Monterrey**

Escuela de ingeniería y ciencias

**Inteligencia Artificial Avanzada para la Ciencia de Datos II
Grupo (501)**

Momento de Retroalimentación: Reto Evaluación

Alumnos:

Jorge Chávez Badillo	A01749448
Ariadna Jocelyn Guzmán Jiménez	A01749373
Juan Carlos Varela Téllez	A01367002
Alan Eduardo Aquino Rosas	A01366912
Amy Murakami Tsutsumi	A01750185

Profesores:

Ivan Mauricio Anaya Contreras
Luis Alberto Muñoz Ubando
Blanca Rosa Ruiz Hernandez
Nora Aguirre-Celis
José Eduardo Ferrer
Felipe Castillo Rendón
Jobish Vallikavungal Devassia

Fecha: 25 de noviembre de 2022

Reto Evaluación

Introducción

Nuestro proyecto consiste en utilizar una implementación de Procesamiento de Lenguaje Natural para la creación de una aplicación web. Esta aplicación tiene como objetivo la facilidad de creación de gráficas y visualización de datos al poder traducir comandos naturales escritos a queries que un graficador pueda entender.

Asimismo, se utiliza un modelo de Aprendizaje Máquina para poder recomendar la mejor gráfica que se pueda representar un conjunto de datos proporcionados.

Esta aplicación dentro de un caso de uso empresarial, podrá ser utilizada por cualquier ejecutivo que requiera sacar conclusiones y tomar decisiones en base a datos que se tiene dentro de su disposición, sin necesidad de tener ningún conocimiento técnico.

Proceso de uso

La interfaz de nuestra aplicación se puede acceder desde cualquier navegador web. Va a contar con 3 pestañas:

- Inicio: en esta pestaña se podrá observar un pequeño resumen del objetivo de la aplicación.



Inicio Crear Dashboard

Inicio

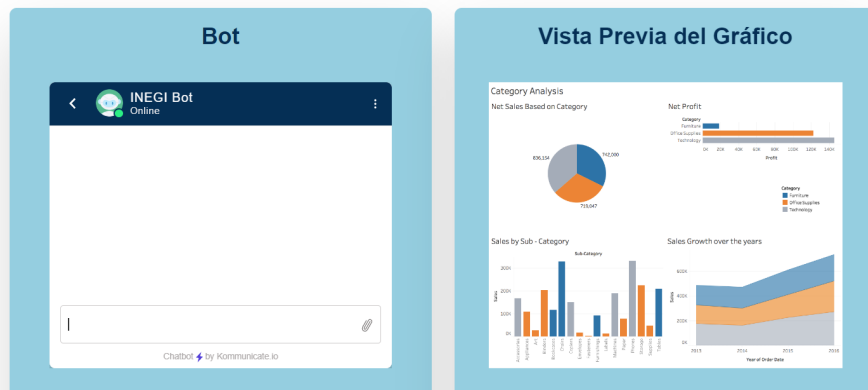


- Crear: esta pestaña cuenta con 2 partes. La primera es la ventana donde se tendrá acceso a un bot que recibirá estos comandos naturales. Una vez que el usuario termine de dar su instrucción, la gráfica resultante de esta instrucción se podrá observar en la segunda parte.



Inicio Crear Dashboard

- Dashboard: por último, este dashboard es el resultado de un análisis que se hizo a

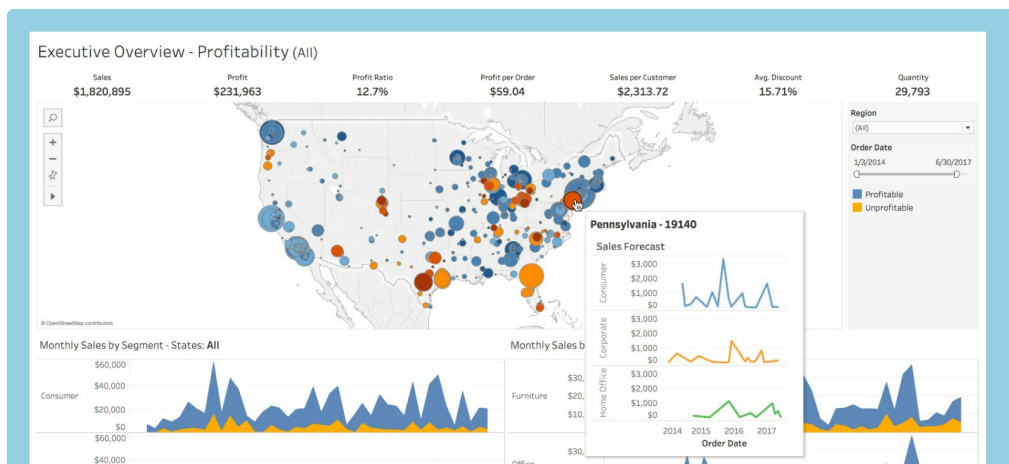


la base de datos que nuestro socio formador INEGI nos proporcionó. Contiene diferentes gráficas que muestran diferentes estadísticas de la población en general.



Inicio Crear **Dashboard**

Dashboard



Tecnologías utilizadas

Servicios de AWS utilizados

Amazon Lex

Amazon Lex es un servicio que implementa Inteligencia Artificial y modelos de Lenguaje Natural Procesado para poder crear bots conversacionales, de esta forma completando las intenciones del usuario para crear cualquier tipo de aplicaciones.

Amazon S3

Este servicio está encargado de almacenar archivos de cualquier tipo. Estos archivos son almacenados en buckets, que básicamente son la unidad lógica del servicio.

En este servicio se ocupará para almacenar el dataset proporcionado por el usuario, para después procesar los datos obtenidos y construir el agente conversacional.

Amazon Lambda

Este es un servicio encargado de ejecutar hooks(bloques) de código en base a un evento proporcionado por uno de los servicios de AWS.

En este caso, las funciones lambda escucharán el evento de subir un dataset/base de datos a un bucket de s3, y preprocesará la información para construir el agente de acuerdo a las necesidades del dataset que se está subiendo.

Amazon Api Gateway

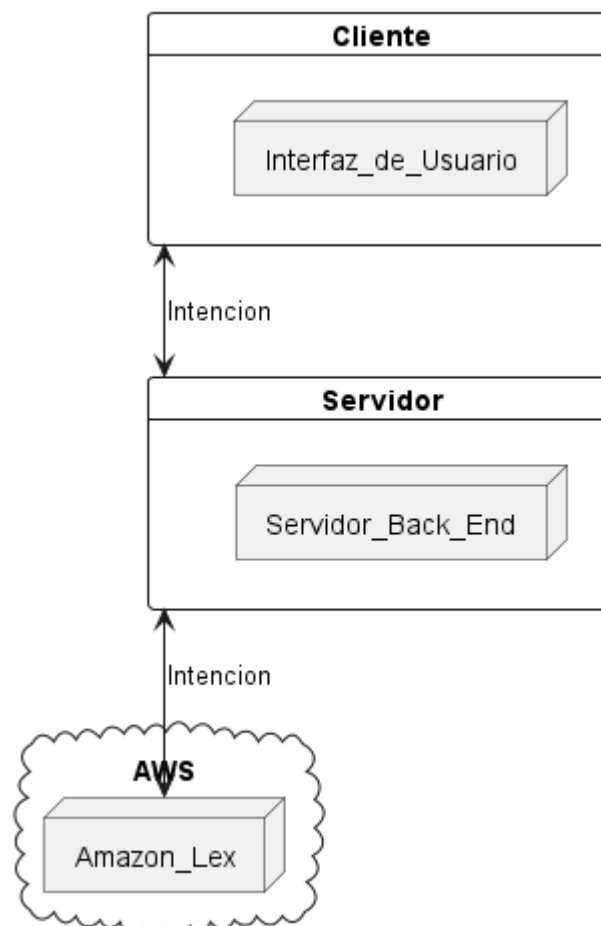
Es un servicio que facilita la implementación de una API utilizando los servicios de AWS. Una API es el puerto de entrada entre la lógica de el backend y otros servicios, así como el front end.

Este servicio se comunicará con el front end(interfaz del usuario) para mandar la información procesada por los servicios de AWS de regreso al cliente. Igualmente, indicará el status de los servicios. Será el orchestrator entre los servicios implementados y el cliente.

Caso de uso: Creación de una gráfica

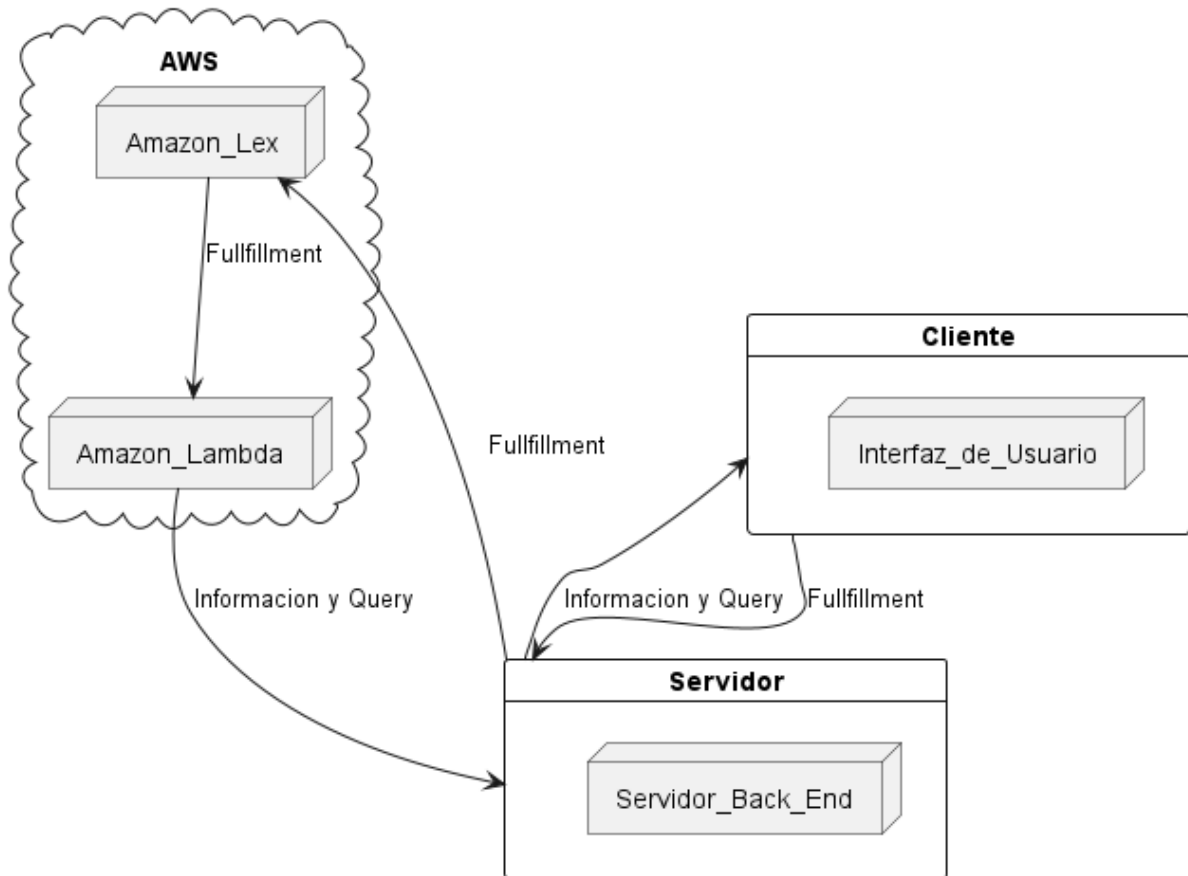
Para poder tener un mejor entendimiento de cómo se crea una gráfica a partir del comando que se le va a dar al bot, es necesario ver detrás de escenas como se mueve la información que se le proporciona a este.

Runtime



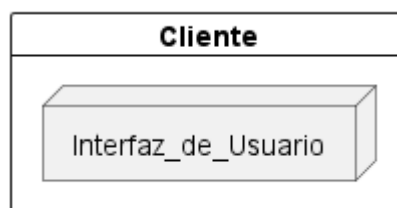
Amazon Lex es el encargado de monitorear el flujo de conversación con el usuario así como saber la intención del mismo. Durante toda la conversación, se hacen varias llamadas API con Amazon Lex para que el bot sepa que responder a lo que pone el usuario y ayudarlo a completar su comando (intención).

Comando Completado



Una vez que el usuario haya completado su comando, Amazon Lex va a activar un trigger y pasará este comando a Amazon Lambda. Este servicio activará una función que filtra la información que el usuario quiere graficar, así como el tipo de gráfica que se va a crear. Amazon Lambda después crea una API que devuelve esta información al servidor y luego al cliente.

Graficar



Por último, con la información necesaria, el mismo cliente con el uso de ChartJS gráfica lo que el usuario pidió en primer lugar, así completando el caso de uso.

Métricas a utilizar

Debido a la naturaleza de nuestro proyecto, no estamos creando un modelo, sino una aplicación que implementa un modelo. Debido a esto, medir el progreso del proyecto se deberá hacer de forma diferente, identificando qué es lo que falta para terminar por completo el proyecto, las cosas que ya se implementan y cómo continuar.











Pruebas del estado

Para poder comenzar con una correcta implementación del bot para tener una experiencia ideal con el usuario, se comenzó con la comprensión de la base de datos a utilizar, en este caso, SCINCE de INEGI, la cual contiene todos los resultados del censo de población realizado en 2020.

A través del entendimiento del contexto, se realizaron consultas sql y operaciones en spark dataframes para poder organizar nuestros datos de acuerdo al estado correspondiente y los elementos que nos parecían más importantes, dando como salida, diferentes archivos csv que posteriormente podrán ser consultados para una búsqueda de información más sencilla.

CVEGEO	POB1	POB31	POB57	POB2	POB4	POB5	POB7	POB11	POB14	POB15	POB16	POB24	ESTADO
1500100010127009	48.0	5.0	0.0	0.0	0.0	0.0	0.0	11.0	17.0	6.0	0.0	5.0	Mexico
1500100010127010	182.0	9.0	10.0	7.0	7.0	15.0	10.0	49.0	43.0	19.0	8.0	24.0	Mexico
1500100010127011	110.0	10.0	9.0	7.0	0.0	9.0	3.0	30.0	21.0	16.0	7.0	15.0	Mexico
1500100010127030	159.0	14.0	11.0	10.0	8.0	12.0	9.0	34.0	46.0	19.0	10.0	11.0	Mexico
1500100010127035	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	3.0	0.0	0.0	Mexico
1500100010269012	13.0	0.0	3.0	0.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0	Mexico
1500100010288007	30.0	0.0	3.0	0.0	0.0	6.0	0.0	7.0	5.0	6.0	0.0	0.0	Mexico
1500100010324004	33.0	0.0	0.0	0.0	0.0	6.0	0.0	10.0	7.0	3.0	0.0	0.0	Mexico
1500100010339008	36.0	4.0	3.0	0.0	4.0	9.0	0.0	5.0	9.0	5.0	0.0	0.0	Mexico
1500100370305001	106.0	7.0	7.0	0.0	5.0	12.0	9.0	29.0	23.0	11.0	6.0	9.0	Mexico
1500100370305002	94.0	5.0	4.0	5.0	6.0	18.0	3.0	18.0	26.0	6.0	3.0	9.0	Mexico
1500100370305009	69.0	0.0	0.0	0.0	5.0	8.0	11.0	22.0	16.0	0.0	0.0	3.0	Mexico
1500100370305011	144.0	8.0	4.0	8.0	10.0	18.0	10.0	36.0	34.0	9.0	5.0	14.0	Mexico
1500100370305017	20.0	3.0	0.0	0.0	0.0	3.0	3.0	0.0	6.0	0.0	0.0	5.0	Mexico
1500100370305018	31.0	0.0	0.0	0.0	0.0	3.0	4.0	6.0	8.0	0.0	0.0	8.0	Mexico
1500100370305033	19.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	3.0	0.0	7.0	Mexico
1500100370305035	105.0	10.0	6.0	6.0	7.0	14.0	6.0	29.0	21.0	11.0	5.0	6.0	Mexico
1500100370305038	79.0	3.0	0.0	0.0	4.0	14.0	0.0	20.0	17.0	5.0	4.0	11.0	Mexico
1500100370305040	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Mexico
1500100370305048	15.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	6.0	0.0	0.0	3.0	Mexico
1500100370305054	24.0	0.0	0.0	0.0	0.0	4.0	0.0	3.0	7.0	0.0	0.0	4.0	Mexico
1500100370305060	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Mexico
150010037031A000	51.0	3.0	0.0	3.0	0.0	7.0	3.0	17.0	16.0	3.0	3.0	9.0	Mexico

ESTADO	POB1	POB31	POB57	POB2	POB4	POB5	POB7	POB11	POB14
Mexico	1.4428917E7	1061146.0	804753.0	526766.0	634979.0	1346058.0	676663.0	3565802.0	419098.0
Jalisco	7364595.0	538462.0	340612.0	289001.0	335147.0	721596.0	332376.0	1845751.0	203530.0
Veracruz	5012679.0	327900.0	276092.0	143041.0	183499.0	441563.0	193020.0	1141976.0	138529.0
Puebla	4951465.0	339855.0	226030.0	207821.0	230030.0	492903.0	229526.0	1278947.0	133454.0
Oaxaca	2440423.0	149426.0	100500.0	79303.0	98628.0	229147.0	98384.0	555700.0	633997.0
CDMX	9138482.0	723590.0	605636.0	235367.0	295056.0	669082.0	339571.0	2089948.0	276027.0
Nuevo Leon	5565047.0	414560.0	264722.0	220019.0	256756.0	513602.0	235580.0	1406286.0	160620.0
Chihuahua	3315288.0	218740.0	146897.0	106230.0	132405.0	309138.0	141791.0	824694.0	918770.0
Guanajuato	4454382.0	318993.0	204197.0	186074.0	209913.0	437517.0	204389.0	1137214.0	123143.0
Michoacan	3380061.0	225026.0	145930.0	138805.0	151817.0	340491.0	144160.0	822482.0	882719.0
Guerrero	2116493.0	126289.0	86282.0	77765.0	85941.0	207472.0	87138.0	498261.0	534822.0
Tamaulipas	3206472.0	213135.0	149254.0	110639.0	130679.0	305144.0	135432.0	767982.0	906057.0
Sonora	2625388.0	165662.0	110218.0	78497.0	95949.0	242904.0	107859.0	638606.0	719169.0
Chiapas	2693999.0	183551.0	104411.0	114737.0	129361.0	292893.0	124431.0	669795.0	720321.0
Baja California	3513335.0	264696.0	156527.0	111436.0	136418.0	321905.0	149037.0	912140.0	105139.0
Coahuila	2903022.0	187792.0	127097.0	122332.0	139464.0	284169.0	120296.0	712351.0	794947.0
Sinaloa	2297899.0	152487.0	107129.0	80771.0	86422.0	212510.0	95040.0	570389.0	627581.0
Yucatan	2034972.0	149521.0	90531.0	67350.0	79949.0	180843.0	80293.0	504908.0	582542.0
Hidalgo	1842583.0	123726.0	85974.0	55527.0	69901.0	171072.0	77215.0	440368.0	521534.0
Queretaro	1873781.0	149007.0	85933.0	66826.0	76857.0	167342.0	75448.0	487861.0	559301.0
Morelos	1591076.0	105368.0	82691.0	51327.0	61363.0	141092.0	65552.0	382028.0	427902.0
San Luis Potosi	1915704.0	132346.0	89814.0	65949.0	78493.0	176148.0	80242.0	481721.0	522219.0
Durango	1336475.0	83643.0	56754.0	51303.0	61117.0	135933.0	55517.0	329335.0	341253.0
Zacatecas	1040268.0	62528.0	39961.0	39228.0	44671.0	103398.0	41169.0	248772.0	265475.0
Tlaxcala	1137399.0	75963.0	51660.0	45979.0	51748.0	113637.0	53717.0	283696.0	304056.0

 poblacion.csv	23 nov 2022	4 KB
 fecundidad.csv	23 nov 2022	2 KB
 mortalidad.csv	23 nov 2022	987 bytes
 migracion.csv	23 nov 2022	2 KB
 hablantes.csv	23 nov 2022	1 KB
 discapacidad.csv	23 nov 2022	4 KB
 educacion.csv	23 nov 2022	3 KB
 economia.csv	23 nov 2022	3 KB
 salud.csv	23 nov 2022	2 KB
 situacion_conyuqal.csv	23 nov 2022	2 KB

Áreas de oportunidad

Al día de hoy (25/11/2022), hace falta la comunicación de los datos filtrados hacia el cliente y la creación de las gráficas.

Para la comunicación de los datos filtrados se está implementando una función lambda que se activa al momento que Amazon Lex diga que el comando está completo. Utilizando las ranuras correspondientes de Amazon Lex, se filtraran columnas del dataset especificado

utilizando la librería pandas de Python. A continuación, se actualizará un API utilizando Amazon API Gateway para que el cliente pueda obtener la información filtrada.

Lo que faltaría es la creación de las gráficas. Como ya se mencionó anteriormente, se utilizará la librería ChartJS para poder graficar estos datos para que se vean visualmente estéticos y tengan una vista profesional. Asimismo, hay funciones prediseñadas que implementan animaciones para estas gráficas.