**Manual de Instalación RAPIDS e implementación en Google Colabority**

1. Entorno de ejecución

Habilitar el entorno de ejecución en colabority, tiene que estar en GPU (Unidad de procesamiento gráfico).





1. Importar tensorflow

Debemos importar el tensorflow para verificar que estamos trabajando con GPU y no tener errores en eventos futuros.

Entrada:

%tensorflow\_version 2.x

import tensorflow as tf

device\_name = tf.test.gpu\_device\_name()

if device\_name != '/device:GPU:0':

  raise SystemError('GPU device not found')

print('Found GPU at: {}'.format(device\_name))

Salida:

Found GPU at: /device:GPU:0

1. Instalación de RAPIDS:

Verificamos con que GPU estamos trabajando, Google Colabority lanza por defecto las siguientes arquitecturas de GPU.

Entrada, código:

!nvidia-smi

Salida:





Tomar en cuenta para que RAPIDS funcione en Google Colabority debe trabajarse con las arquitecturas de GPU siguientes:

* Tesla T4
* Tesla P4
* Tesla P100

Instalación:

# Install RAPIDS

!git clone https://github.com/rapidsai/rapidsai-csp-utils.git

!bash rapidsai-csp-utils/colab/rapids-colab.sh

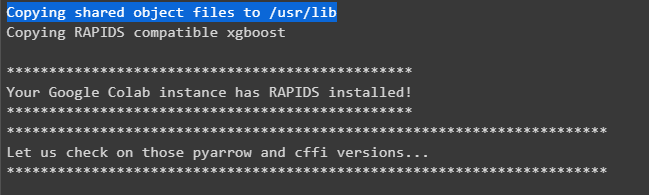
import sys, os

dist\_package\_index = sys.path.index('/usr/local/lib/python3.6/dist-packages')

sys.path = sys.path[:dist\_package\_index] + ['/usr/local/lib/python3.6/site-packages'] + sys.path[dist\_package\_index:]

sys.path

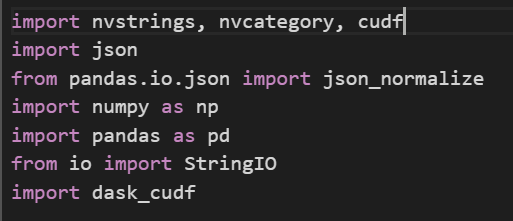
exec(open('rapidsai-csp-utils/colab/update\_modules.py').read(), globals())



Si la instalación no tuve errores, se ejecutó de manera correcta quiere decir que han seleccionado el paquete estable de RAPIDS que es la versión 0.13.

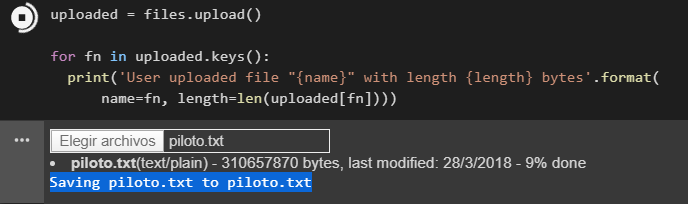
1. Importar librerías de RAPIDS:

**Tomar en cuenta:** Importar los paquetes de nvstrings y nvcategory antes de cuDF, si no se importa estos paquetes la biblioteca cuDF no funcionara, ejemplo:



1. Carga de Archivo a Google Colabority.

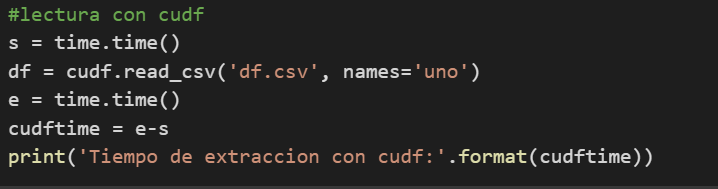
Cargar archivo.txt o cualquier otro archivo desde la memoria local del computador personal a Google colabority, para después guardar el archivo en la memoria temporal de colabority:



1. Trabajo con cuDF.

Dato: cuDF tiene la misma sintaxis que la librería PANDAS (se parece en mucho, pero cuDF incorporar características muy importantes)

La siguiente figura corresponde a la lectura del archivo con las características de cuDF.

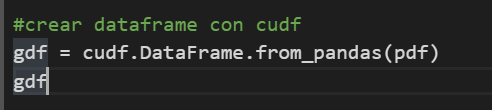


cudf.read\_csv: lee archivos CSV.

cudf.read\_json: lee archivos json.

cudf.read\_txt: lee archivos de texto.

1. Convertir un DataFrame.



cudf.DataFrame.from\_pandas: convierte un DataFrame de Pandas a un nuevo DataFrame cuDF.

cudf.DataFrame.from\_records: convierte un DataFrame de numpy a un nuevo DataFrame cuDF

1. DataFrame creado y funcionalidades del DataFrame con cuDF.

gdf.head(): imprime las 5 primeras filas del DataFrame

gdf.dtypes(): imprime el tipo de datos de cada columna del dataframe

gdf.columns(): imprime el nombre de las columnas del dataframe

gdf.iloc(): permite seleccionar las filas e imprime las filas seleccionadas del dataframe

gdf.ndim(): imprime la dimensión del dataframe

gdf.shape(): imprime la representación del dataframe

gdf.describe(): imprime una descripción del dataframe por columnas que tengan valores numéricos

gdf.drop(‘nombre\_columna’): elimina una columna

gdf.drop\_duplicates (‘nombre\_columna’): elimina duplicados de una columna

gdf.fillna(): rellena valores nulos

gdf.join(): permite unir DataFrames