

Trabajo Práctico N°3

Miño Maximiliano, Otalora Griselda, Gnecco Cristian
41783275, 39213045, 40024360
Miércoles, Grupo 7

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina

Resumen. En este informe se explicará la implementación de algoritmos de computación de altas prestaciones (HPC). Por un lado se implementó el algoritmo de Box Blur, el cual fue realizado mediante el enfoque de procesamiento secuencial(CPU) y paralelo usando la GPU. Por otra parte se realizó la visualización en gráficos de cálculos realizados para ciertos casos y pronósticos del Covid. Ambos desarrollados en Colab, del cual se explicará por que se lo utilizo y que se necesita para trabajar en él. Además, se proveerá un manual de usuario de cada cuaderno para el entendimiento de ambos algoritmos.

Palabras claves: Cuda, Python, Box Blur, Colab, Covid, HPC, GPGPU

1 Introducción

En el siguiente informe se hablara de como fuimos desarrollando el algoritmo Box Blur[1][2] y la investigación e implementación de visualizaciones en gráficos de distintos casos del Covid respecto al tiempo y un pronóstico de posibles casos, que se desarrollaron en Colab, aplicando Python y Cuda. Primero vamos a explicar que es Colab y por qué se lo utilizó. Colab es una Plataforma online de Google, que te permite ejecutar y programar en Python de manera online y fue utilizado, en principio, porque es un requisito para la entrega del TP, pero además el uso de Colab nos brinda un ambiente con las siguientes prestaciones sin necesidad de instalar nada: GPGPU-nvidia, SDK CUDA, Visual studio SDK, Python y el Modulo de Jupyter. Además de que se puede trabajar en equipo. Finalmente, para poder utilizar Colab, solo se necesita una cuenta de Google.

Para el ejercicio 1 nos tocó el algoritmo Box Blur, el cual se encarga de hacer un efecto de difuminado en una imagen y esto lo hace tomando cada pixel de la imagen, obteniendo el promedio de los píxeles que lo rodean(incluyendo al mismo pixel). Este algoritmo se desarrolló mediante dos enfoques, el de procesamiento secuencial (CPU) y el procesamiento paralelo, usando la GPU. Más adelante, en la parte de Conclusiones del ejercicio 1 se realiza una explicación de los resultados

obtenidos con respecto al tiempo al momento de comparar el desarrollo en CPU y GPU.






Para el ejercicio 2 se realizó el análisis y visualización de datos de un dataset con información de casos confirmados, muertes y recuperados del Covid-19 y se realizó un pronóstico para los siguientes meses. Para esto se utilizó módulos y bibliotecas como: Pandas, Matplotlib y fbprophet.

2 Desarrollo

Ejercicio 1:

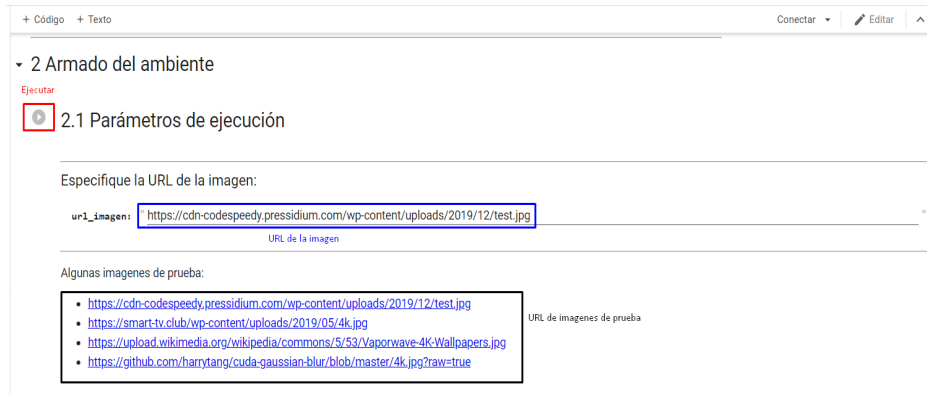
a. Explicar detalladamente como utilizar el ejercicio (Manual de usuario)

El cuaderno cuenta con un índice del lado izquierdo en la parte superior, donde se encontrarán las siguientes secciones:

| | | |
|---|---|---|
|  | Índice |  |
|  | 1 Introducción | |
| | 2 Armado del ambiente | |
|  | 2.1 Parámetros de ejecución | |
|  | 2.2 Instalacion en el cuaderno del módulo CUDA de Python. | |
| | 3 Desarrollo CPU | |
| | 4 Desarrollo GPU | |
| | 5 Metricas | |
| | 6 Conclusiones | |
| | 7 Bibliografia | |

Para la ejecución del ejercicio, primero, se debe ir a la sección “Armado de ambiente”

Y elegir la subsección “2.1 Parámetros de ejecución”, donde se debe insertar la URL de la imagen a la que se le aplicará el algoritmo, para que esté cargado en el cuaderno y luego debe ejecutar dicha sección.



También dispondrá de 4 enlaces a imágenes de pruebas, que fueron las utilizadas para la sección 5 (Métricas).

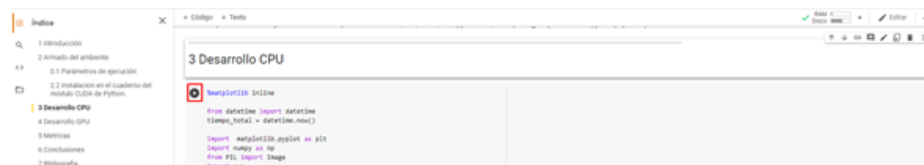
El siguiente paso es ir a la subsección “2.2 Instalación en el cuaderno del módulo CUDA de Python”, el cual también debe ser ejecutado, para que se instale el módulo de CUDA en el cuaderno y poder ejecutar la sección 4 “Desarrollo GPU”.



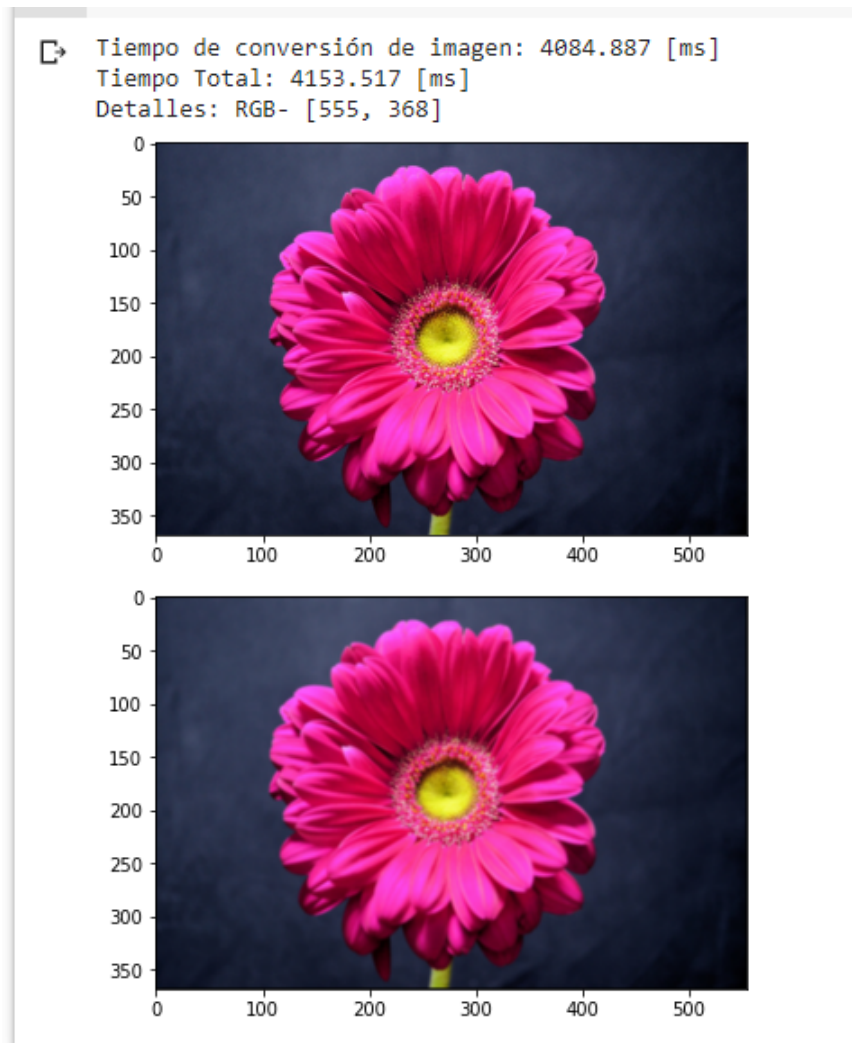
Una vez finalizado el armado del ambiente, recién puede pasar a usar el algoritmo desarrollado, donde puede elegir entre el desarrollado en CPU o con la GPU.

| | | |
|----|---|---|
| ☰ | Índice | ✕ |
| 🔍 | 1 Introducción | |
| | 2 Armado del ambiente | |
| <> | 2.1 Parámetros de ejecución | |
| 📁 | 2.2 Instalacion en el cuaderno del módulo CUDA de Python. | |
| | 3 Desarrollo CPU | |
| | 4 Desarrollo GPU | |
| | 5 Métricas | |
| | 6 Conclusiones | |
| | 7 Bibliografía | |

En ambos casos, haciendo clic en dicha sección, el cuaderno le mostrará el código del algoritmo desarrollado. Dando la opción de ejecutarlo.



Una vez ejecutado, se debe ir a la parte inferior de dicha sección, donde podrá ver la imagen original y otra imagen que tiene el algoritmo aplicado.



b. Repositorio GitHub

Enlace repositorio GitHub: <https://github.com/crisgnecco/TP2-android>
[Rama: main]

c. Funcionamiento del ejercicio desde el punto de vista de la programación CPU-GPU

La primera parte de ambos desarrollos es similar, ya que se trata de las importaciones de las clases necesarias y la obtención de las medidas de la

imagen a procesar (alto y ancho). En el Desarrollo de la CPU, el algoritmo se trabaja anidando tres for, donde el primer for es el encargado de manejar los tres canales de colores (rojo, verde y azul), y los dos for siguientes serán los encargados de recorrer el ancho y largo de la imagen, para que de esta manera se pueda recorrer toda la imagen, a excepción de sus bordes, y calcular el promedio de los pixeles que lo rodean (incluyendo al mismo pixel).

Mientras que con GPU, en principio se trabaja con bloques de hilos, por lo que se tuvo que definir la cantidad de hilos y bloques, para obtener la dimensión del bloque y de la grilla, esto se lo hizo en base al tamaño de la imagen, de la siguiente manera:

```
dim_hilo_x = 16
```

```
dim_bloque_x = np.int( (img_ancho+dim_hilo_x-1) / dim_hilo_x )
```

```
dim_hilo_y = 19
```

```
dim_bloque_y = np.int( (img_alto+dim_hilo_y-1) / dim_hilo_y ).
```

Luego se procedió a cargar el código cuda, en el cual se calculan las coordenadas de los hilos en dos dimensiones, para aprovechar la multidimensión y así poder trabajar con toda la imagen. En esta función es donde se realiza el promedio del píxel, sumando todos los que tiene alrededor y el mismo.

Para poder trabajar con los tres colores de cada píxel, se realiza un for con tres canales y dentro de este for, es llamado el kernel para poder transformar la imagen.

d. ¿Qué sucede si se planifican la mitad de los hilos-gpu?

Si la cantidad de hilos se reduce a la mitad, el tamaño del bloque también se reduce a la mitad y se duplica la cantidad total de bloques a planificar pero se observó que la variación del tiempo no sufrió un cambio significativo en los tiempos de ejecución tanto en la parte de GPU como en el TOTAL.

Planificación de hilos original

Tiempo TOTAL: 40.028 [ms]

Tiempo GPU : 0.239 [ms]

Planificación de hilos mitad

Tiempo TOTAL: 39.694 [ms]

Tiempo GPU : 0.285 [ms]

¿Qué sucede si se planifican el doble de hilos-gpu?

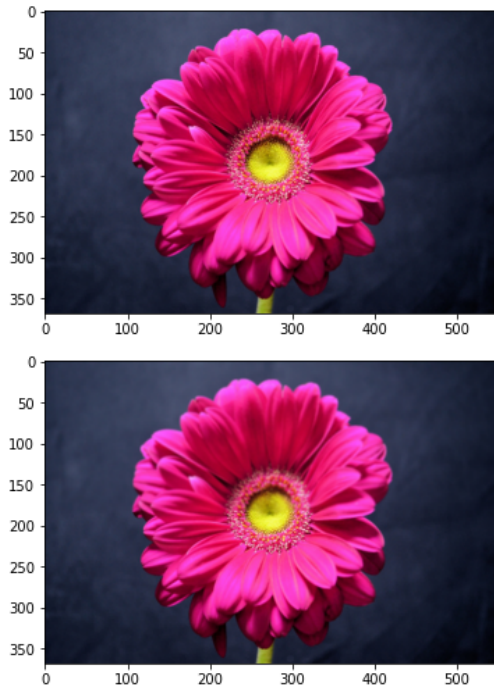
En este caso, se obtuvo un error en el argumento dimensión de bloque que se le pasa al kernel debido a excedernos en la cantidad de hilos por bloque

(tamaño de bloque). Estamos intentando crear bloques de 32 por 38 hilos lo cual, excede al máximo de 1024 hilos por bloque.

¿Qué sucede si solo se procesa la mitad inicial de la imagen original?

Al procesar la primera mitad de la imagen original en alto (se dividió el alto de la imagen a la mitad), se obtiene la mitad de la imagen procesada en la salida con el mismo efecto que ingresando la original, y la otra mitad se encuentra sin el efecto aplicado.

```
➤ Detalles: RGB- [555, 184.0]  
Tiempo TOTAL: 37.263 [ms]  
Tiempo GPU : 0.202 [ms]
```



¿Qué sucede si solo se debe procesar la mitad final de la imagen destino?

Al procesar solamente la segunda mitad de la imagen final en alto (se dividió dentro del IF, donde se hace la verificación de que los hilos se encuentren dentro del tamaño de la imagen), se obtiene la mitad de la imagen con el

mismo efecto, mientras que la primera mitad, se encuentra con su forma original.

```
➡ Detalles: RGB- [555, 368]  
Tiempo TOTAL: 37.388 [ms]  
Tiempo GPU : 0.2 [ms]
```



¿Qué sucede si se utilizan bloques con la cantidad máxima de hilos soportada por el GPGPU?

La cantidad máxima de hilos por bloque es 1024 para la GPU asignada, en nuestro caso es la arquitectura Tesla 4.

Si utilizamos la cantidad máxima de hilos soportados. Es decir, bloques de 32 por 32 hilos, obtenemos un resultado satisfactorio y no observamos variaciones significativas con respecto a la ejecución con 16 por 19 hilos.

Ejercicio 2:

a. Conceptos aplicados

El concepto que se está aplicando es volumen de datos:

En la primera parte graficamos la cantidad de casos, muertes o recuperados en función del tiempo según la entrada del usuario. Como entrada tenemos una tabla con 279 filas que corresponden a la cantidad de casos de cada día por cada país y 540 columnas, de las cuales 536 corresponden a las fechas.

A tener en cuenta: El dataset se actualiza cada día, por lo tanto la cantidad de columnas aumentará en 1 cada día.

En la segunda parte usamos una biblioteca llamada profet (internamente usa la biblioteca sci learn) la cual nos permite pronosticar la cantidad de casos para los próximos meses. Esto se logra entrenando al algoritmo con nuestro dataset para generar un modelo predictivo y luego se obtiene pronóstico para un periodo dado.

Para hacer esta predicción se requiere una entrada de datos de gran volumen si bien no es un factor directamente proporcional a la precisión del pronóstico.

b. Manual de usuario

El cuaderno cuenta con un índice del lado izquierdo en la parte superior, donde se encontrarán las siguientes secciones:

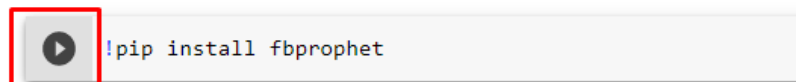
| | | |
|----|--|---|
| ☰ | Índice | ✕ |
| 🔍 | 1 Introduccion | |
| | 2 Armado del ambiente | |
| <> | 2.1 Instalacion del modulo fbprophet | |
| 📁 | 2.2 Importacion de modulos y datasets necesarios | |
| | 2.3 Declaracion de funciones necesarias para el Desarrollo | |
| | 3 Desarrollo | |
| | 3.1 Grafico con respecto al tiempo | |
| | 3.2 Pronostico de casos | |
| | 4 Conclusiones | |
| | 5 Bibliografia | |

Para la ejecución de la sección “Desarrollo”, primero, se debe ir a la sección “Armado de ambiente” y ejecutar las 3 subsecciones que contiene.

Primero debe ir a la subdirección “2.1 Instalación del módulo fbprophet” y ejecutarlo.

▼ 2.1 Instalacion del modulo fbprophet


Ejecutar



El siguiente paso es ir a la subsección “2.2 Importación de módulos y datasets necesarios” y ejecutarlo.

▼ 2.2 Importacion de modulos y datasets necesarios

Ejecutar

```
 %matplotlib inline

# Importando paquetes necesarios
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sys
```

Por último ejecutar la subsección “2.3 Declaración de funciones necesarias para el desarrollo”.

▼ 2.3 Declaracion de funciones necesarias para el Desarrollo

Ejecutar

```
 # Listado de todos los paises del dataset
lista_paises = list(data_confirmados['Country/Region'].drop_duplicates())

# Funciones

# Funcion para agrupar por pais
def agrupar(df):
    # Agrupo por pais y elimino las columnas "Lat" y "Long"
    data = df.groupby(['Country/Region']).sum().drop(['Lat', 'Long'], axis=1)
    # Realizo transposicion del dataset
    data = data.transpose()
    # Seteo los indices (Fechas) como DateTimeIndex
    datetime_index = pd.DateTimeIndex(data.index)
    data.set_index(datetime_index, inplace=True)
    return data
```

Una vez finalizado el armado del ambiente, recién puede pasar a la sección “Desarrollo”, donde podrá optar entre las subsecciones:

3 Desarrollo


3.1 Grafico con respecto al tiempo

3.2 Pronostico de casos

En la subsección “3.1 Gráfico con respecto al tiempo” debe ejecutar la sección:

▼ 3.1 Grafico con respecto al tiempo

Ejecutar

```
 paiseseleccionados = []  
  
@interact  
def graficar(Pais = ['...']+lista_paises,  
  
             global paiseseleccionados
```

y se encontrará con lo siguiente en la parte inferior:

 Pais

☐ Mundial

Filtro

Esto permitirá visualizar un gráfico interactivo en la parte inferior dependiendo de lo que seleccione:

En la lista “País” podrá ir agregando países a su gráfico de la siguiente manera:

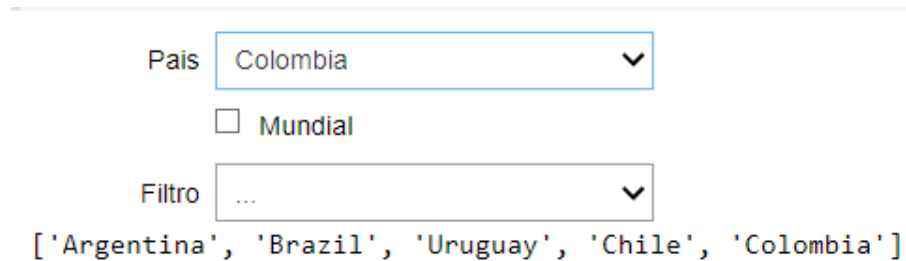
Pais

Antigua and Barbuda
Argentina
Armenia
Australia
Austria

Filtro

Repetir esta selección tantas veces como países desee en el gráfico.

En la parte inferior se informará de la lista de países seleccionado



País Colombia ▼

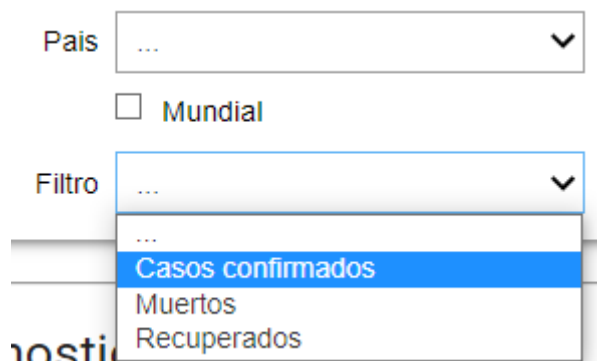
☐ Mundial

Filtro ... ▼

`['Argentina', 'Brazil', 'Uruguay', 'Chile', 'Colombia']`

Al seleccionar la opción “Mundial” permite agregar todos los países al gráfico. Si quita la opción, volverá a graficar los países anteriormente seleccionados en la lista “País”

En la lista “Filtro” podrá seleccionar con respecto a que filtró desea el gráfico. Estos son: Casos Confirmados, Muertes y Recuperados.



País ... ▼

☐ Mundial

Filtro ... ▼

- ...
- Casos confirmados
- Muertos
- Recuperados

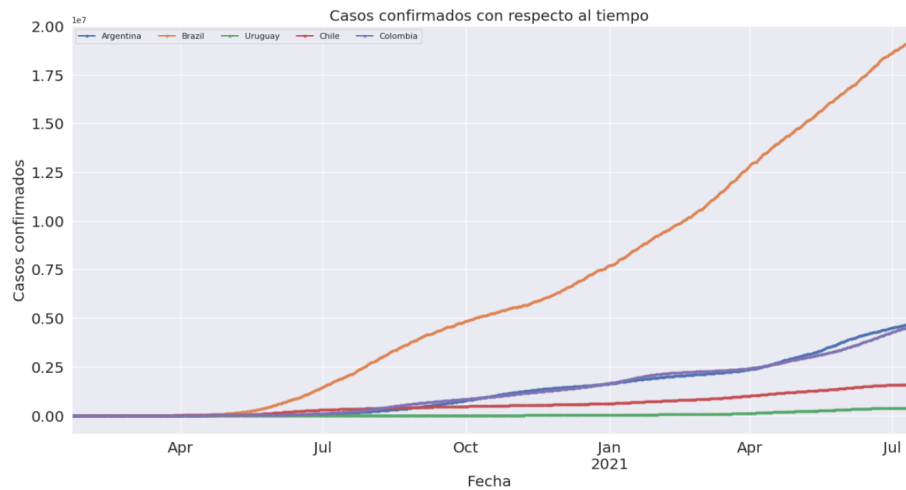
Tener en cuenta que para que se pueda graficar debe tener seleccionado algún país y algún filtro, o en su defecto tener seleccionada la opción “Mundial” y un filtro.

Si desea eliminar todos los países del gráfico y empezar de nuevo debe volver a ejecutar la sección.

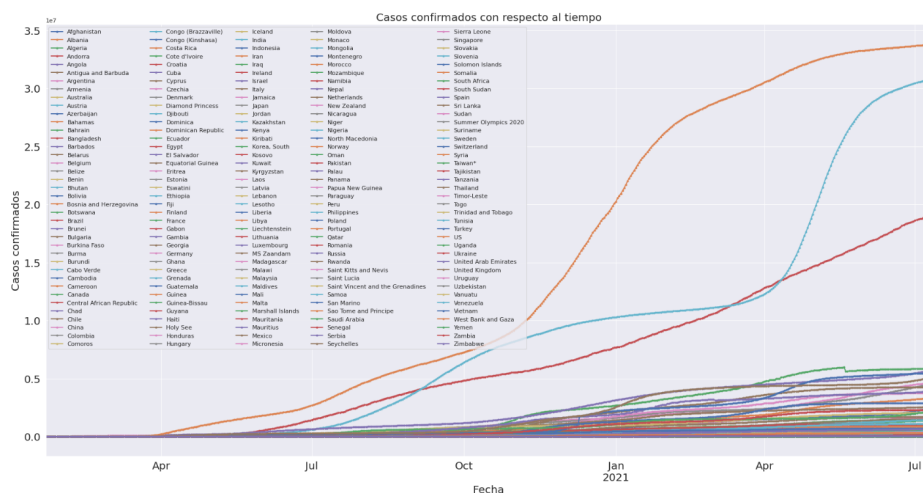
En la parte inferior, se podrá visualizar un gráfico de la cantidad de recuperados, muertos o casos confirmados (dependiendo que se seleccionó en la subsección) en función del tiempo de todos los países seleccionados

Algunos ejemplos son:

Si se selecciona los países: Argentina, Brazil, Uruguay, Chile y Colombia y el filtro “Casos Confirmados”:



Si seleccionó el filtro “Casos confirmados” y la opción “Mundial”:



Por otra parte, en la subsección “3.2 Pronóstico de casos”, se muestra un gráfico de los posibles futuros contagios, muertes o recuperados en el país seleccionado o en su defecto, en el mundo, de acuerdo a los datos obtenidos del dataSet .

Al ejecutar la subdirección se encontrará con las mismas opciones a elegir que en la subdirección 3.1. Pero la diferencia es que solo se podrá elegir un país, o en su defecto todos los países de dataset seleccionando la opción “Mundial” y que encontrara un deslizable el cual corresponde al periodo al cual se desea pronosticar a futuro, que va desde 1 a 12 meses

Pais

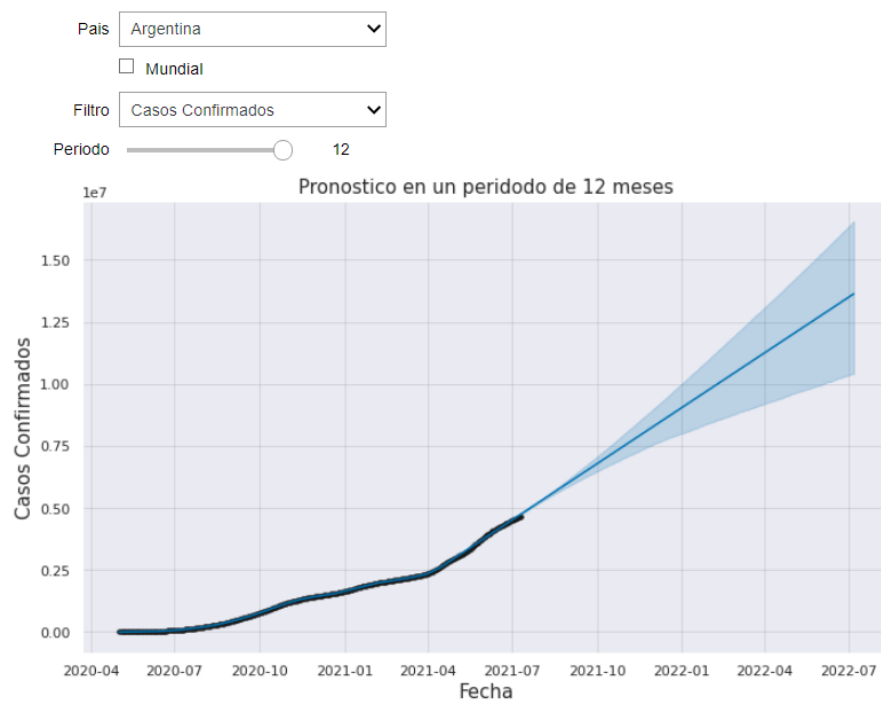
☐ Mundial

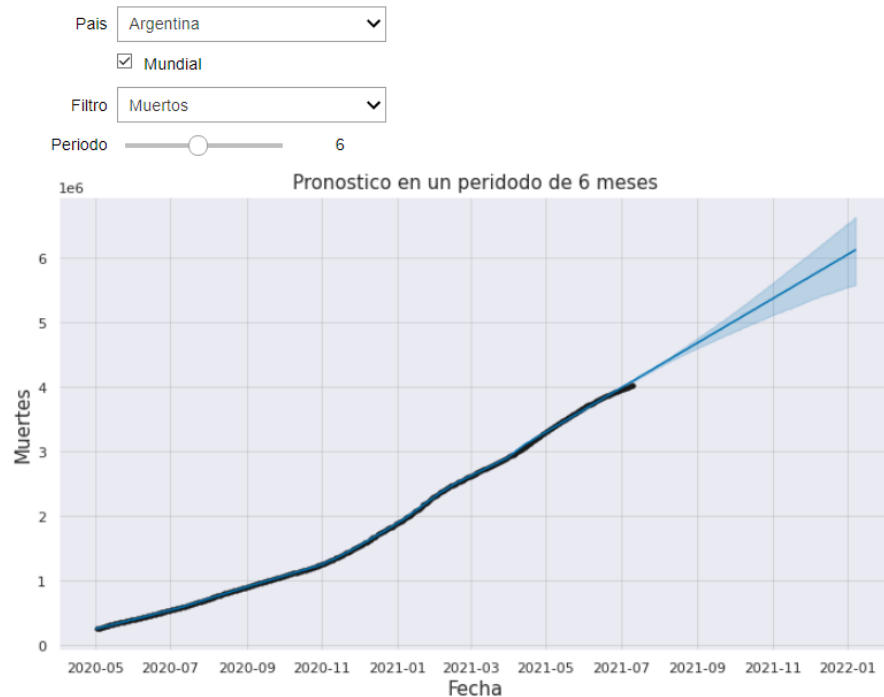
Filtro

Periodo 12

Tener en cuenta que para que se pueda graficar debe tener seleccionado algún país y algún filtro, o en su defecto tener seleccionada la opción “Mundial” y un filtro.

Algunos ejemplos son:





c. Repositorio GitHub

Enlace repositorio GitHub: <https://github.com/crisgnecco/TP2-android>
[Rama: main]

3 Conclusiones

Problemas surgidos

Para el ejercicio 1 tuvimos problemas con los canales RGB de la imagen, hasta que investigamos sobre esto y entendimos que la imagen se compone de tres elementos: `imagen[x,y,canal]` y que el canal 0 es rojo, el canal 1 verde y el canal 2 azul.

Para el ejercicio 2 tuvimos problemas al seleccionar el tema, pero se solucionó eligiendo el tema del Covid por ser algo muy actual y de gran utilidad para la sociedad y que se aplica al tema HPC al contar con un gran volumen de datos en los datasets elegidos.

Con el código no tuvimos mayor problema, más allá del poco conocimiento

en análisis y visualización de datos, y en python y sus módulos, al elegir este lenguaje de programación se simplificó la tarea, al tener una gran abstracción frente al programador y al poseer tanta documentación y foros abiertos en internet al cual consultar.

Ejercicio 1

Con la resolución de este ejercicio mediante los enfoques de procesamiento secuencial (CPU) y paralelo usando la GPU queda en evidencia la diferencia en tiempo de ambas soluciones.

Se pone en evidencia que a menor sea las dimensiones de las imágenes usadas los tiempos obtenidos de ambas soluciones se asemejan y hasta la solución paralela llega a ser peor en cuestión de tiempos. Por el contrario, a mayor sea las dimensiones de las imágenes los tiempos obtenidos en la solución paralela son abismalmente mejores que los tiempos de la solución secuencial.

Ejercicio 2

La conclusión que se llega al realizar este ejercicio es del poder de python y sus módulos para el análisis y la visualización de datos. Además de la utilidad de la plataforma Colab y la confección de cuadernos para demostrar nuestro trabajo.

En este caso se eligió el tema Covid, y se realizó el análisis de la visualización de los datos de casos, muertes y recuperados de 3 datasets con gran volumen de datos. Este seguimiento actualizado todos los días sirve tanto a nivel político, económico y social para tomar decisión de acuerdo a los datos obtenidos de análisis de estos datasets.

Por último el ejercicio nos sirvió para conocer un poco más en detalle los módulos de python pandas y matplotlib. Y conocer el módulo Prophet para el pronóstico de casos futuros y su gran potencial.

4 Referencias

- [1] «Geeksforgeeks, Box Blur Algoritm» 2021. [En línea]. Available: <https://www.geeksforgeeks.org/box-blur-algorithm-with-python-implementation/>
- [2] « Scholarworks, Different Approaches to Blurring Digital Images and Their Effect on Facial Detection » 2019. [En línea]. Available: <https://scholarworks.uark.edu/cgi/viewcontent.cgi?article=1067&context=csceuht>

- [3] <https://colab.research.google.com/github/hugobowne/COVID-19-EDA-tutorial/blob/master/notebooks/1-COVID-19-EDA-solution.ipynb#scrollTo=4LdokSt3aIYE>
- [4] « Pandas.Pydata, Pandas.DataFrame.groupby » 2021. [En línea]. Available: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html>
- [5] « Matplotlib, Matplotlib: Visualization with Python » 2021. [En línea]. Available: <https://matplotlib.org>
- [6] « Facebook.github, Quick Start – Python API » 2021. [En línea]. Available: https://facebook.github.io/prophet/docs/quick_start.html#python-api
- [7] « Medium, Prediction of the number of Covid-19 cases using Google Colab, Python and Facebook's Prophet library » 2021. [En línea]. Available: <https://medium.com/@simonprdhm/prediction-of-the-number-of-covid-19-cases-using-google-colab-python-and-facebooks-prophet-732caee170f8>
- [8] «Researchgate, Time Series Facebook Prophet Model and Python for COVID-19 Outbreak Prediction » 2021. [En línea]. Available: https://www.researchgate.net/publication/349833934_Time_Series_Facebook_Prophet_Model_and_Python_for_COVID-19_Outbreak_Prediction