

# Calidad del Aire

+ Lluvia ácida

Monóxido de carbono +

Óxidos de nitrógeno +

+ Ozono

+ Dióxido de azufre

+ Partículas suspendidas



+ Plomo

Proyecto Final de Deep Learning

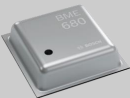

Presentan:

Jorge III Altamirano Astorga,  
Luz Aurora Hernández Martínez,  
Ita-Andehui Santiago Castillejos.

# Introducción: Fuente de datos

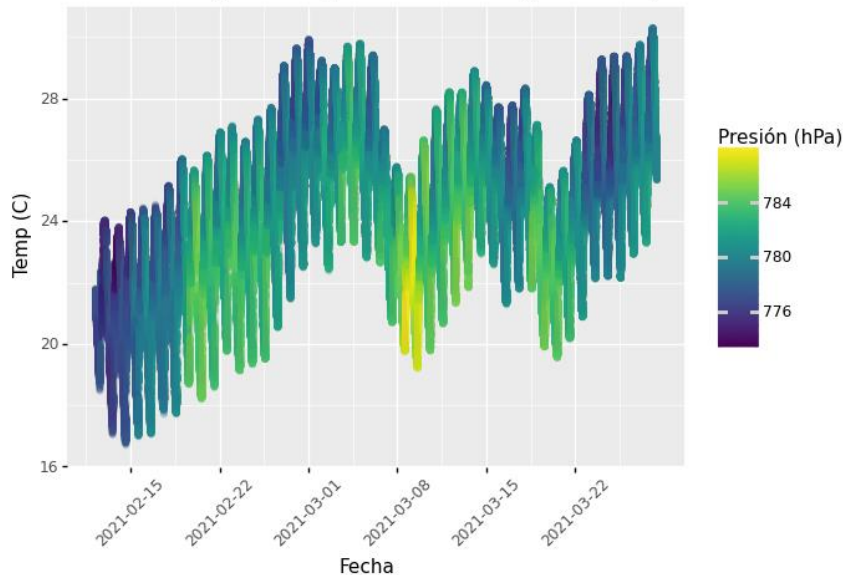
| Fuente  | Descripción  | Registros   | Resolución  |
|---|--|-------------|---|
|   | Sensor Bosch para medir contaminantes en interior.           | +2 Millones | <ul style="list-style-type: none"> <li>• Cada 3 segundos</li> </ul> |
|  | Datos del Gobierno de las Estaciones de Monitoreo Ambiental. | +2,100      | Cada 60 segundos  |

# Introducción: Variables.

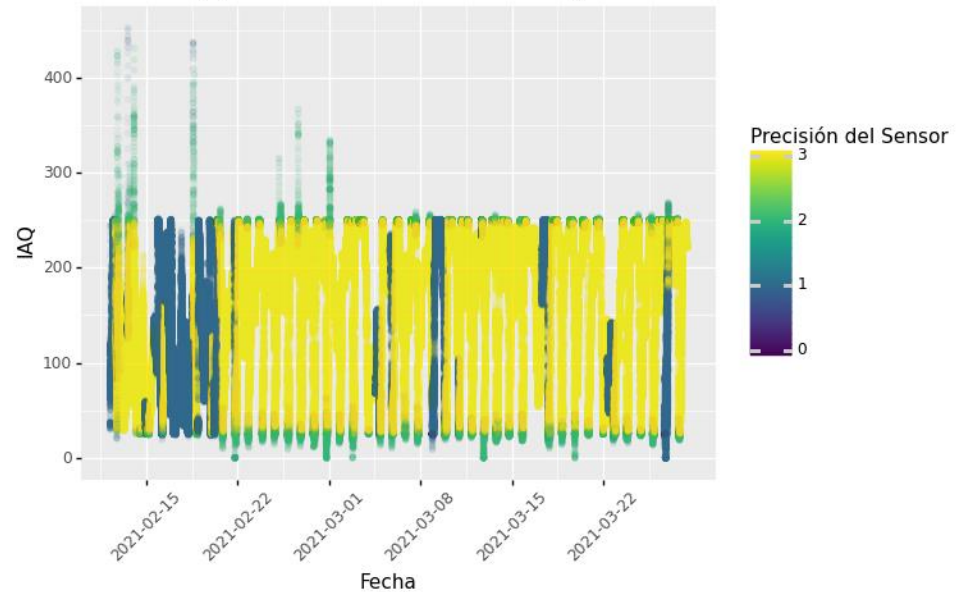
|     | Fuentes   | Variable            | Rango de Valores        | Tipo de Variable |
|-----|---|---------------------|-------------------------|------------------|
| $x$ |    | Temperatura         | -40C a 85C              | Continua         |
|     |    | Humedad             | 10% a 95%               | Continua         |
|     |    | Presión Atmosférica | 300 hPa - 1100 hPa      | Continua         |
|     |   | Fechas y Hora       | 12/02/2021 - 24/04/2021 | *                |
|     |   | Contaminantes       | ppm principalmente      | Discreta         |
| $y$ |    | Resistencia del Gas | 0 Ohms - 3 Mega Ohms    | Continua         |
|     |    | IAQ                 | 0 IAQ - 500 IAQ         | Continua         |

# Introducción: Exploración de Datos del Sensor

Gráfica de Temperatura y Presión a lo Largo del Tiempo.

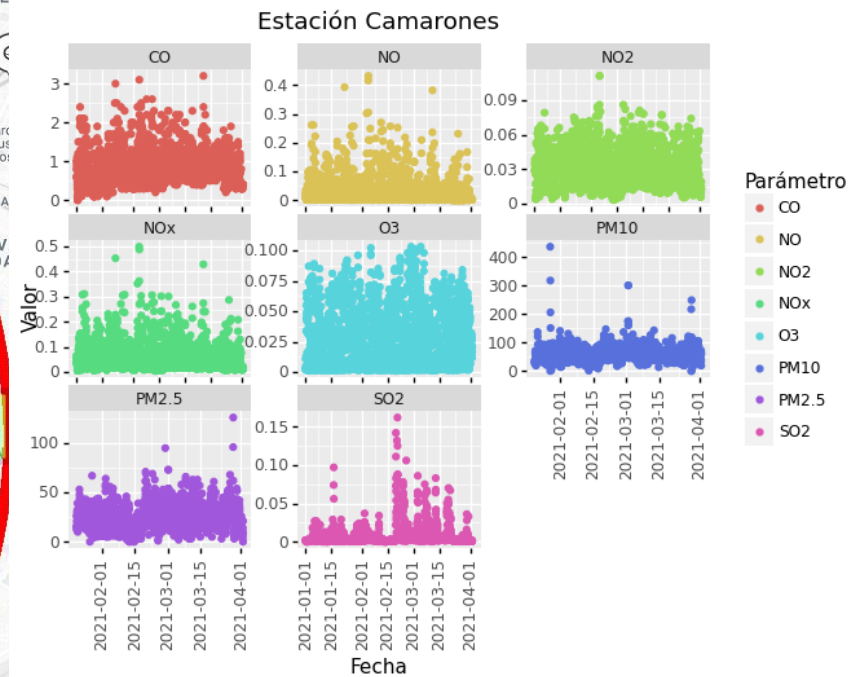


Gráfica de IAQ y Precisión del Sensor a lo Largo del Tiempo



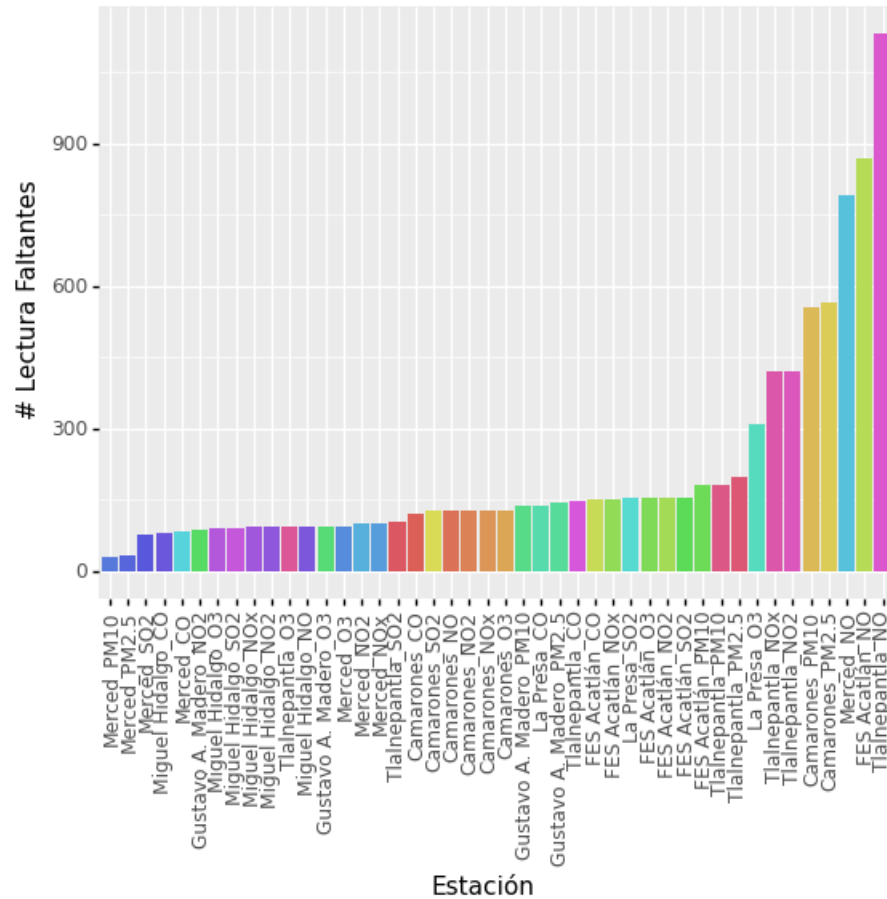
Datos faltantes: ~1%

# Introducción: Exploración de Datos SINAICA

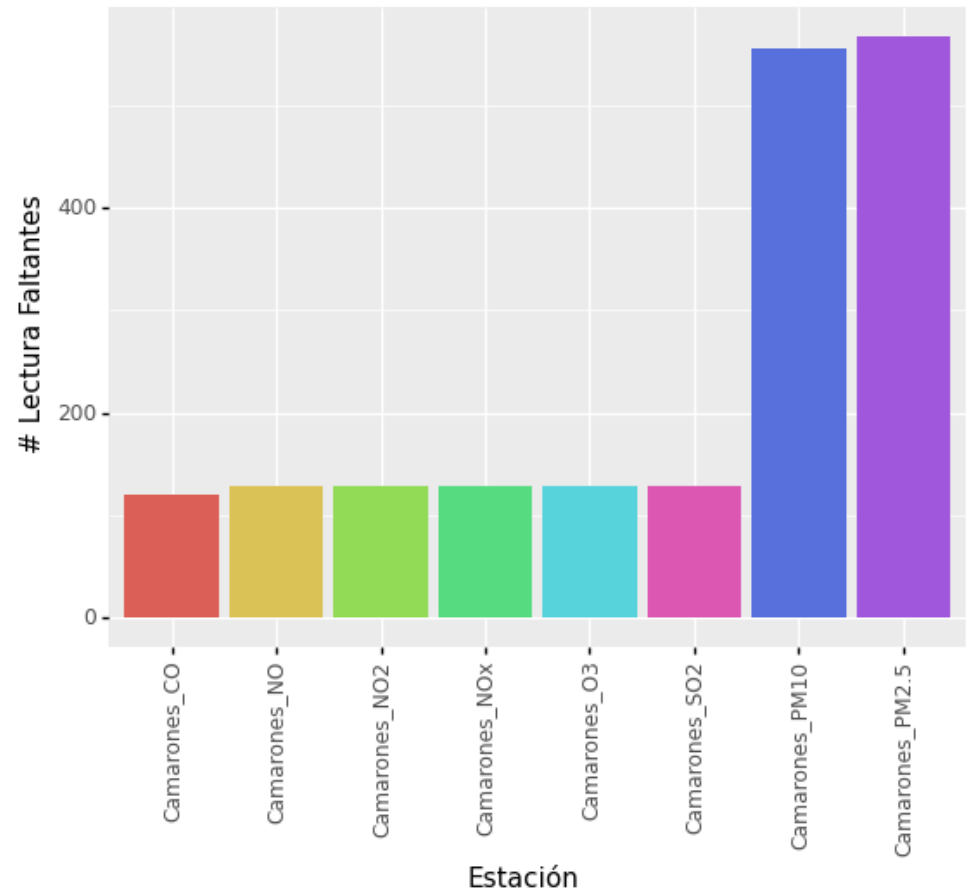


# Introducción: Exploración de Datos del Gobierno

Histograma de Lecturas Faltantes por Contaminante-Estación de Monitoreo

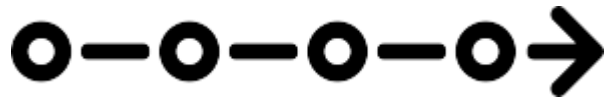


Histograma de Lecturas Faltantes por Contaminante en la Estación Camarones





# Solución: Preprocesamiento



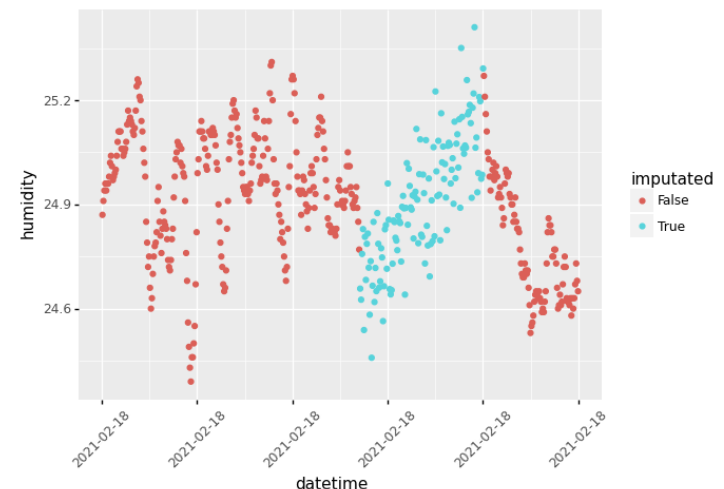
Procesamos los datos como una Serie de Tiempo: como en el Miniproyecto 4 y en un tutorial oficial de Tensorflow y Keras.



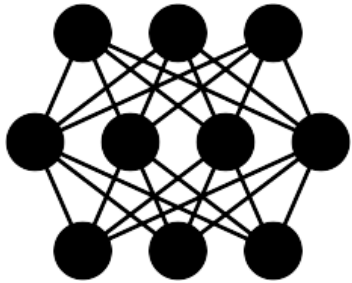
Tuvimos que imputar, porque **todos** los datos tenían algún faltante, como se vió anteriormente. Usamos interpolación, aunque exploramos KNN, Métodos Lineales Generalizados (Bayes), Medias, Hot Deck.

Escalamiento: al tener datos en diversas escalas.

Limpieza de Datos: descartar primeras observaciones por el Windowing.

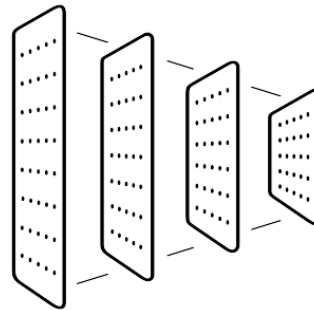
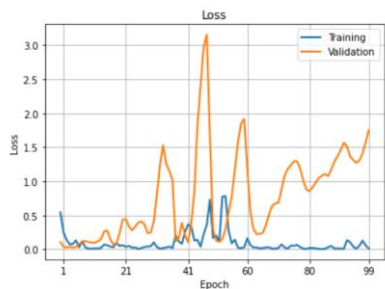


# Solución: Arquitectura de Redes Neuronales



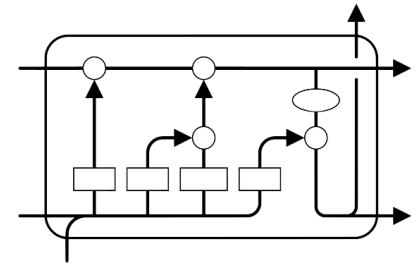
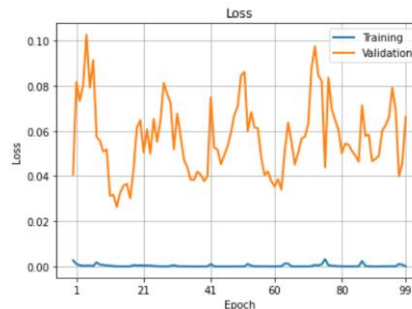
## Dense:

- Simple y Rápida.
- No entregó tan buenos resultados.
- Imprescindible:  
Es la base del resto de los distintas arquitecturas.



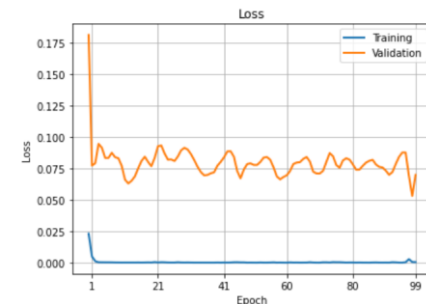
## Convolutional 1D:

- Desempeño robusto.
- Demandante en procesamiento.
- Resultados “ruidosos”.



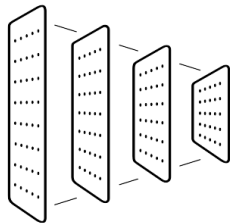
## LSTM:

- Desempeño razonable.
- Procesamiento intermedio.
- Resultados estables.

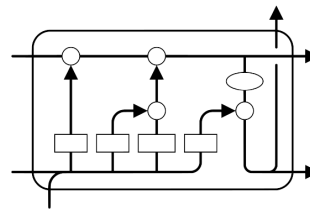




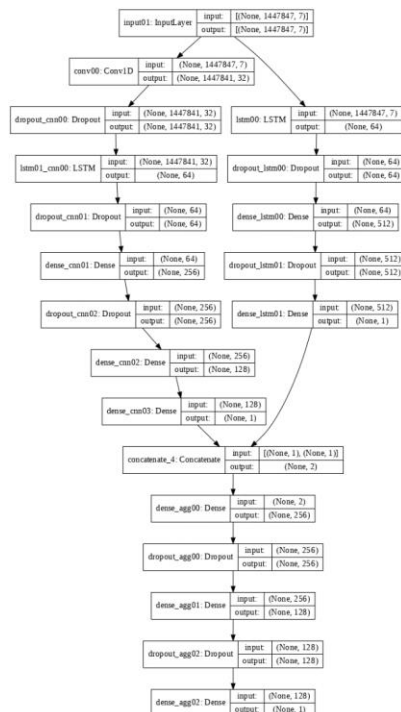
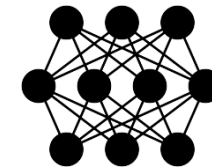
# Solución: Propusimos Combinar CNN+LSTM+DNN



+



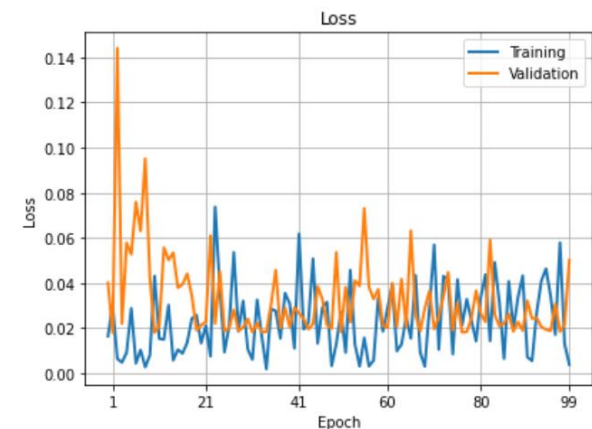
+



## Combinación de Redes:

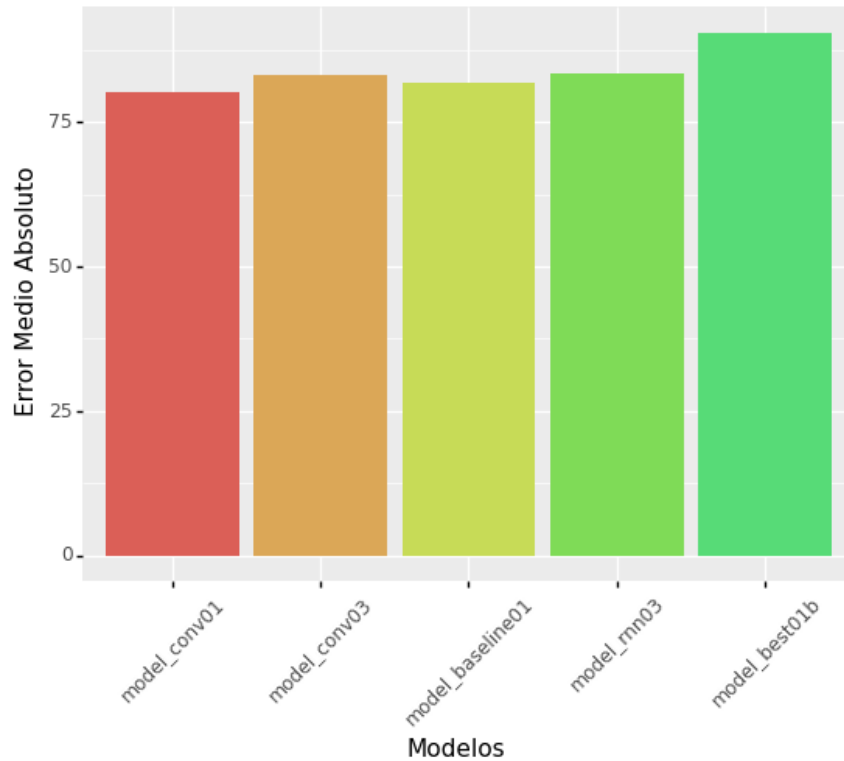
- *Esperábamos resultados sustancialmente mejores.*
- *Logramos desempeño estable y razonable.*
- *El tiempo de entrenamiento fue bastante razonable, aún teniendo una arquitectura compleja.*
- *Técnicamente fue un reto implementarlo.*

| Modelo        | Tiempo   | # Params | val_mae | mae   |
|---------------|----------|----------|---------|-------|
| model_conv01  | 14m3.57s | 294,401  | 62.76   | 62.73 |
| model_best03b | 8m50.63s | 162,115  | 74.06   | 61.15 |
| model_best03a | 14m0.58s | 485,633  | 75.94   | 55.80 |
| model_lstm03  | 9m11.55s | 183,297  | 87.76   | 21.71 |

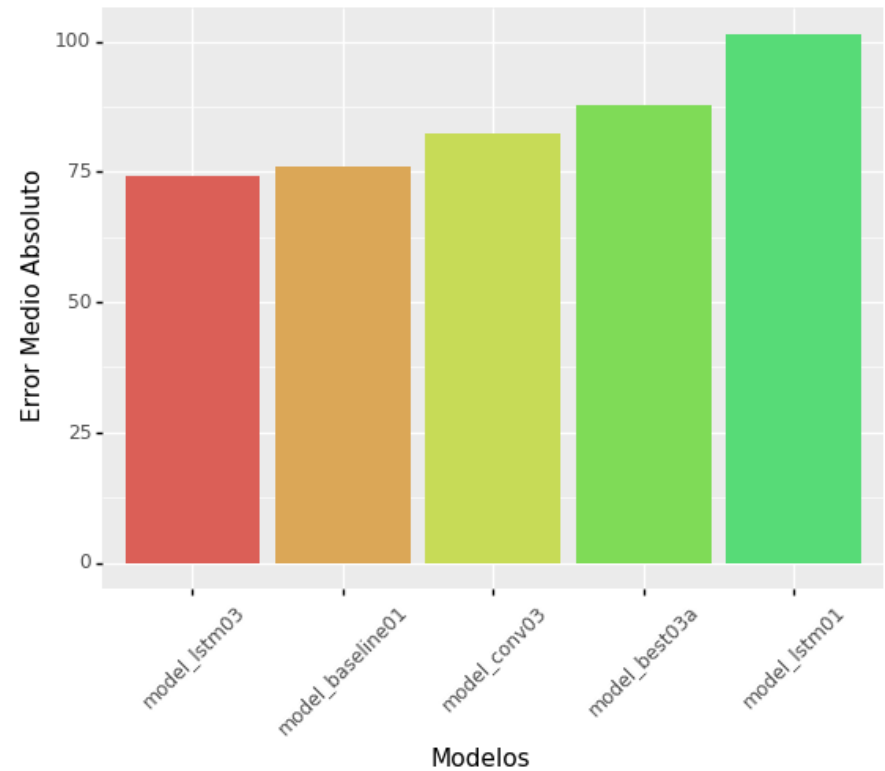


# Resultados

Gráfica Comparativa entre los Modelos con Datos del Gobierno y el Sensor



Gráfica Comparativa entre los Modelos con Datos del Sensor



# Conclusiones: Logros y Siguietes Pasos

- Logros:
  - Logramos poder predecir y es medible el desempeño modelo.
  - Logramos reducir el sobreajuste.
  - Logramos aprender sobre la realización de un proyecto *end-to-end*, sobre redes neuronales y las series de tiempo.
- Siguietes Pasos:
  - Hacer modelos más grandes y con más historia.
  - Buscar cómo mejorar el desempeño con *hyper parameter tuning* y la arquitectura de la red.
  - Modificar la forma de tratamiento de las series de tiempo.

# Conclusiones: Aprendizajes

- ✓ Cumplir con los principios científicos: reproducibilidad y repetibilidad.
- ✓ Nunca se debe subestimar la inversión de tiempo necesaria para limpiar, explorar, imputar, “corregir” y conocer los datos.
- ✓ ¡Mejorar el desempeño es difícil!
- ✓ No se debe confiar en la disponibilidad de datos externos.
- ✓ Hay muchísimos recursos en Internet: buenos y malos.
- ✓ Las APIs cambian: No tener miedo a aprender continuamente.
- ✓ Nos resultó muy útil tener un modelo *baseline*: nuestra  $H_0$
- ✓ Tener cuidado con los detalles.
- ✓ “Des-escalar” los datos nos dio una idea más clara del desempeño.
- ✓ Es efectivo ir construyendo de modelos simples → modelos más elaborados. También probar, probar, probar.

# ¡Gracias!

*¿Preguntas?*