

# BRAIN TUMOR CLASSIFICATION

**Matías Werbin - Noelia Qualindi**

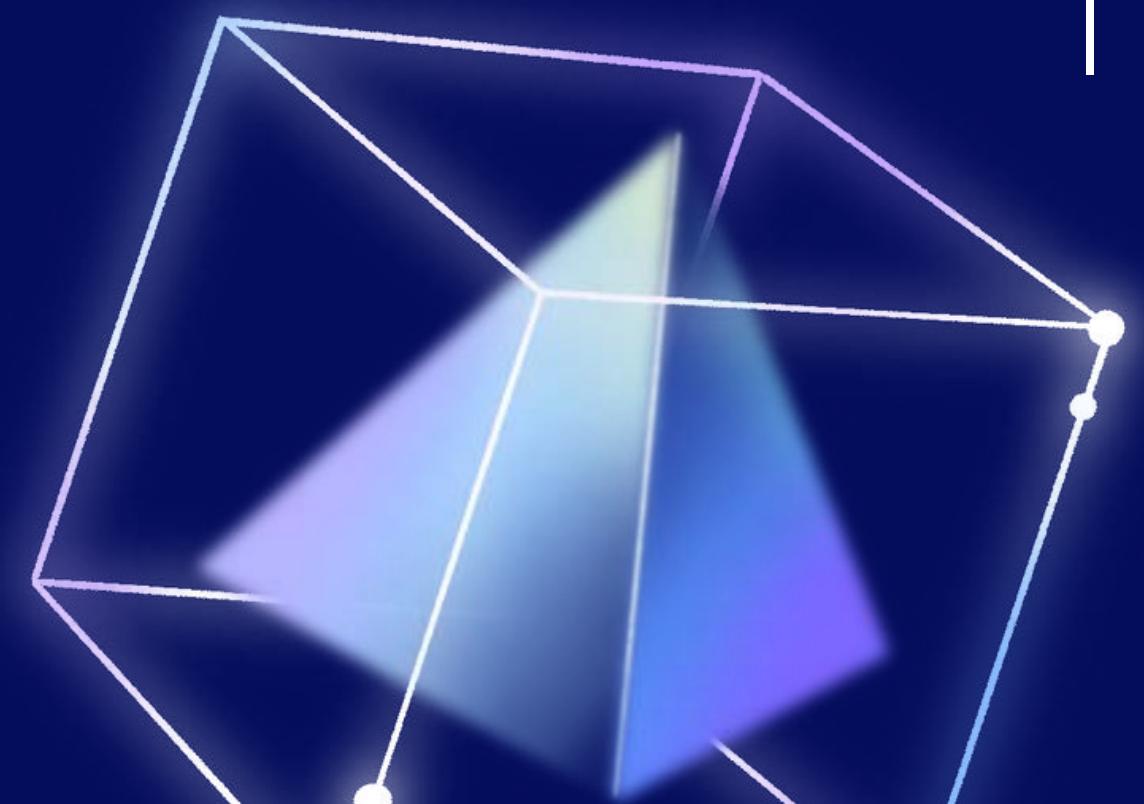
Cohorte 14 - Año 2024

VISION POR COMPUTADORA II

Docentes:  
Juan Ignacio Cornet  
Juan Ignacio Cavalieri

# TABLA DE CONTENIDO

- Introducción
- Análisis exploratorio
- Arquitecturas y enfoques
- Performance
- Resultados y Logros
- Conclusión
- Demo



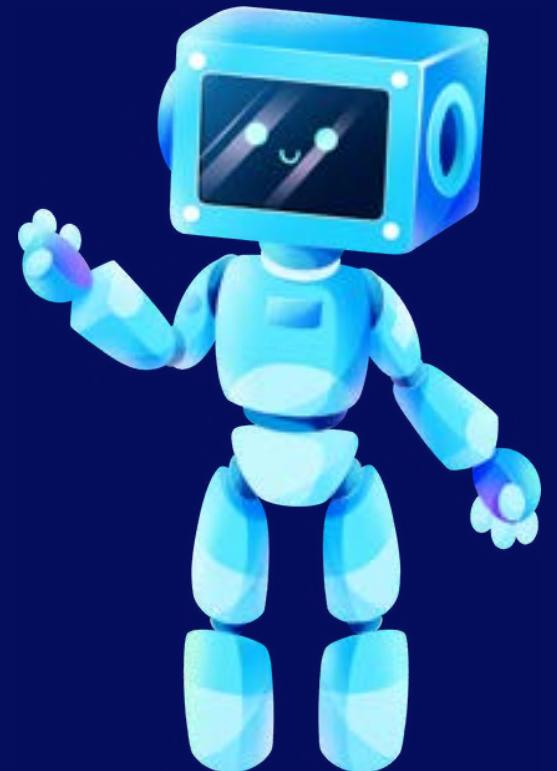
# INTRODUCCIÓN

La detección temprana y precisa de tumores cerebrales es crucial para mejorar el pronóstico de los pacientes, pero sigue siendo un desafío en la medicina. Por ello, es urgente desarrollar sistemas automatizados basados en inteligencia artificial que clasifiquen eficientemente imágenes médicas para identificar tumores cerebrales de manera rápida.

---

## SOLUCIÓN PROPUESTA

En este proyecto, exploraremos un enfoque basado en redes neuronales profundas pre entrenadas para clasificar imágenes cerebrales y distinguir entre tres tipos de tumores (Gliomas, Meningiomas y tumores Pituitarios) como así también identificar la ausencia de tumor.



# ANÁLISIS EXPLORATORIO DE DATOS

DATASET: BRAIN TUMOR CLASSIFICATION

EL DATASET ESTÁ DIVIDIDO EN DOS  
CARPETAS: TRAIN Y TEST

```
# Crear conjuntos de datos
X_train, y_train = create_dataset(TRAIN_PATH, CATEGORIES, IMG_SIZE)
X_test, y_test = create_dataset(TEST_PATH, CATEGORIES, IMG_SIZE)

print(f"X_train shape: {X_train.shape}")
print(f"y_train shape: {y_train.shape}")
print(f"X_test shape: {X_test.shape}")
print(f"y_test shape: {y_test.shape}")

100%|██████████| 826/826 [00:00<00:00, 1297.45it/s]
100%|██████████| 822/822 [00:00<00:00, 1415.36it/s]
100%|██████████| 395/395 [00:00<00:00, 1482.05it/s]
100%|██████████| 827/827 [00:00<00:00, 1232.50it/s]
100%|██████████| 100/100 [00:00<00:00, 1135.49it/s]
100%|██████████| 115/115 [00:00<00:00, 1525.40it/s]
100%|██████████| 105/105 [00:00<00:00, 2527.56it/s]
100%|██████████| 74/74 [00:00<00:00, 623.50it/s]
X_train shape: (2870, 224, 224, 3)
y_train shape: (2870,)
X_test shape: (394, 224, 224, 3)
y_test shape: (394,)
```

TRAIN

**2870** muestras de  
entrenamiento

TEST

**394** muestras para test

# ANÁLISIS EXPLORATORIO DE DATOS

DATASET: BRAIN TUMOR CLASSIFICATION

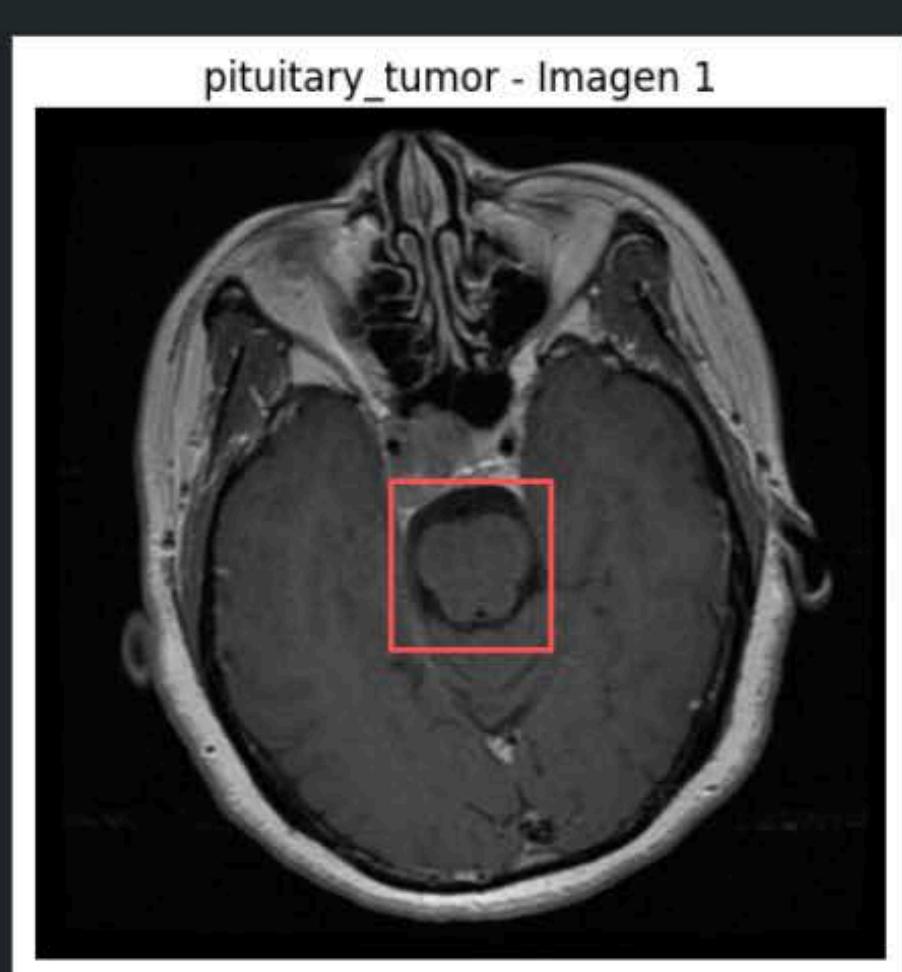
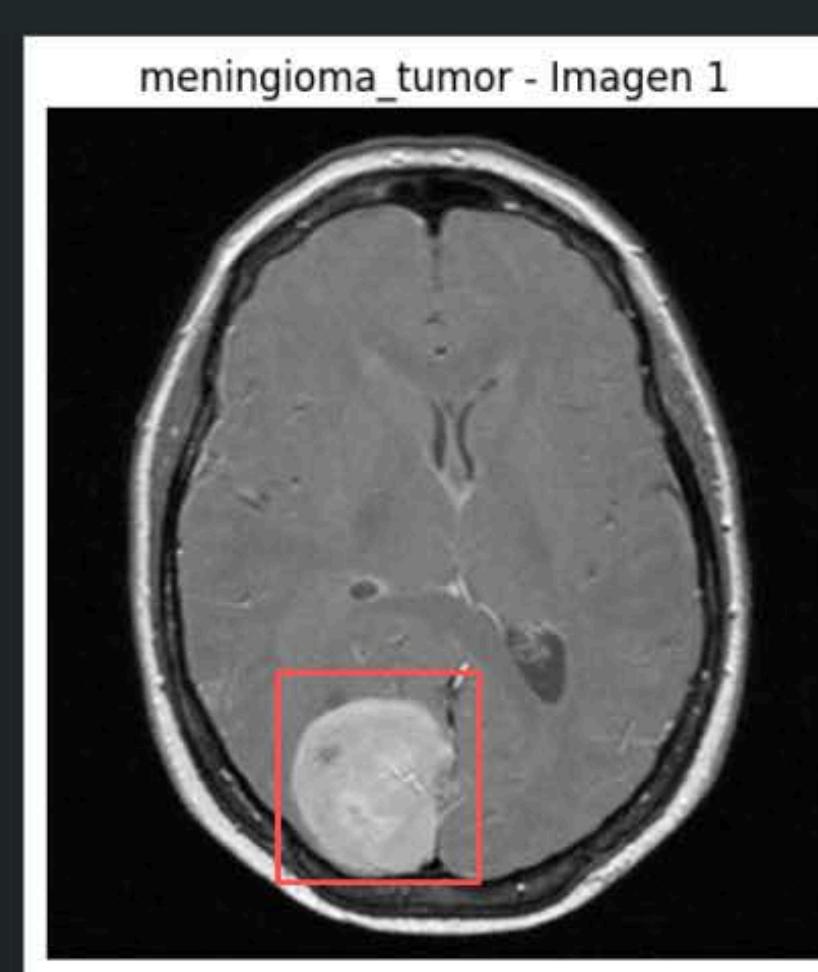
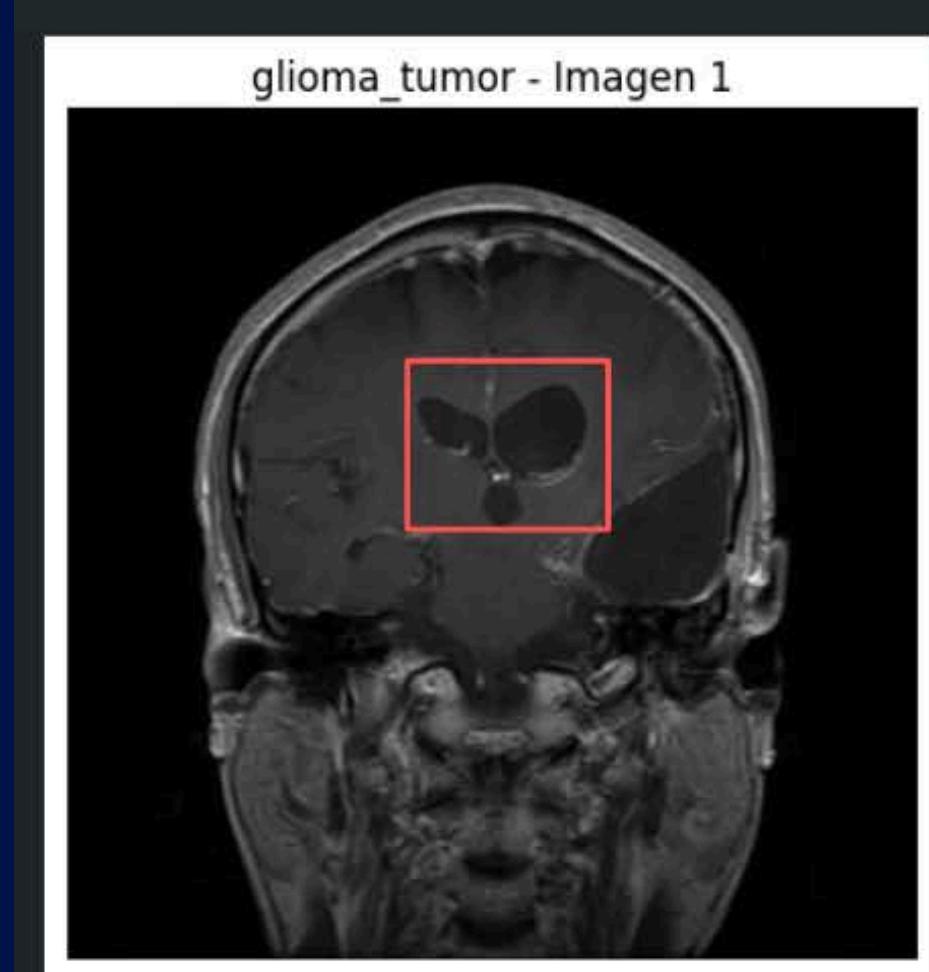
4 CLASES DE SALIDA

glioma\_tumor

meningioma\_tumor

no\_tumor

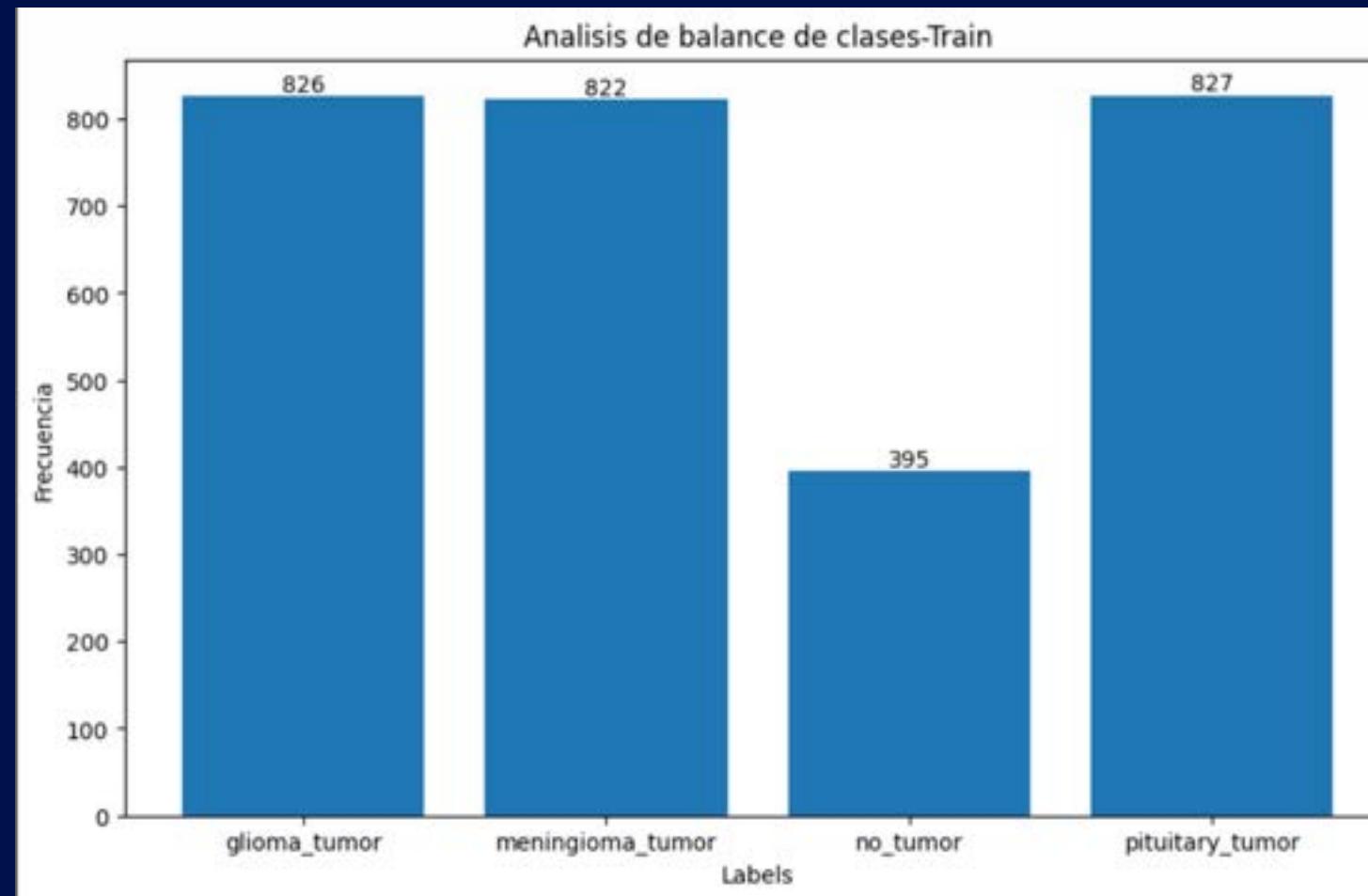
pituitary\_tumor



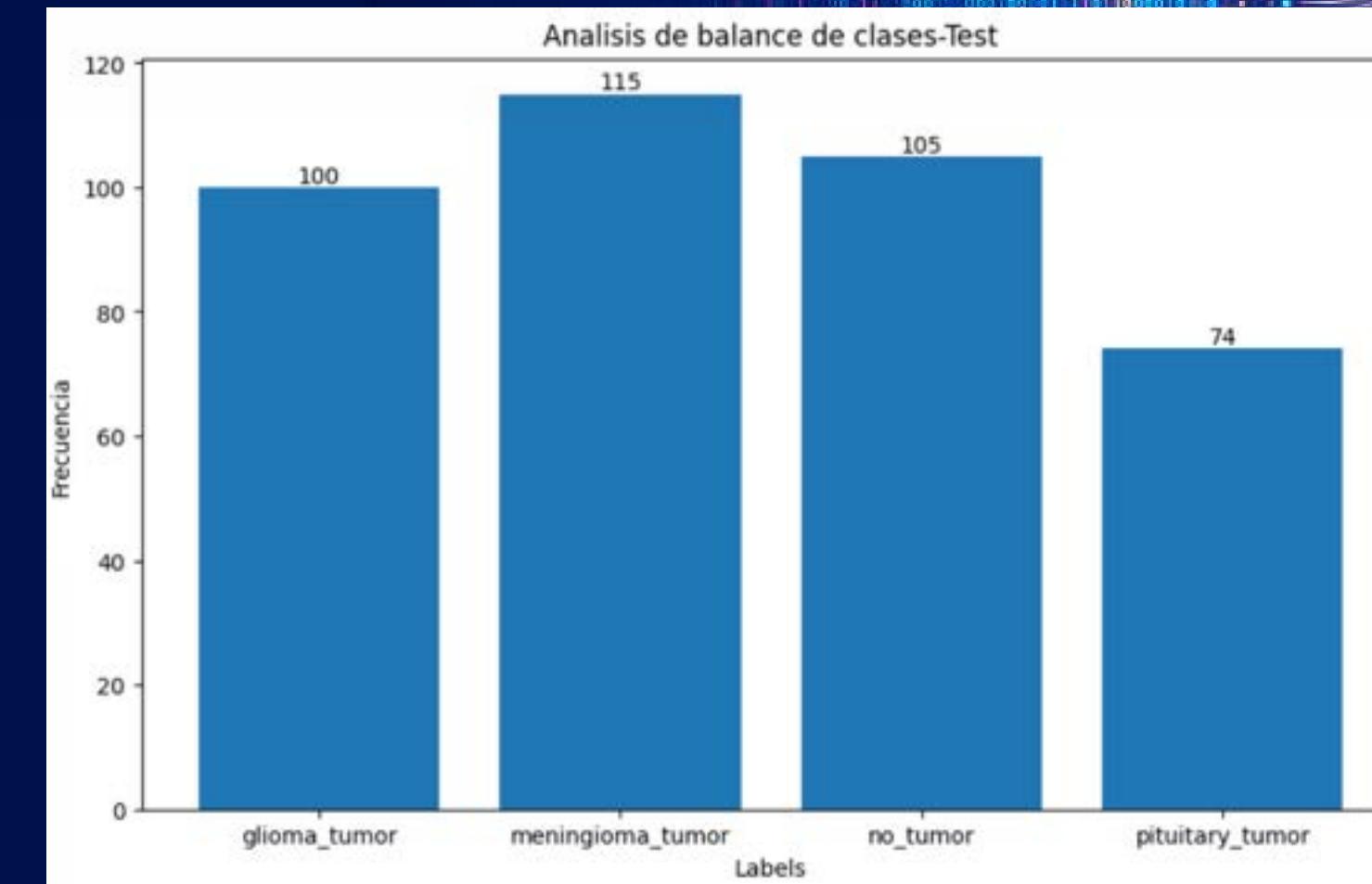
# ANÁLISIS EXPLORATORIO DE DATOS

## MÉTRICA DE MONITOREO Y EVALUACIÓN: ACCURACY

### BALANCE DE CLASES



El set de train presenta un desbalance de clase. En este caso, la clase minoritaria es no\_tumor. A priori no debería presentar un problema porque nuestro objetivo principal es que la red pueda aprender a clasificar las imágenes que son tumores.



El set de testeо se encuentra bastante balanceado.

SE PUEDE CONSIDERAR A MODO GENERAL QUE LOS CONJUNTOS SE ENCUENTRAN BALANCEADOS



# ARQUITECTURAS Y ENFOQUES UTILIZADOS

---



01

## TRANSFER LEARNING NO DATA AUGMENTATION

- **VGG16 | ADAM | LR: 0.001**
  - Feature Extraction
  - Complete Fine Tuning
- **Resnet 50 | ADAM | LR: 0.001**
  - Feature Extraction
  - Complete Fine Tuning
- **Inception-V3 | ADAM | LR: 0.001**
  - Feature Extraction
  - Complete Fine Tuning

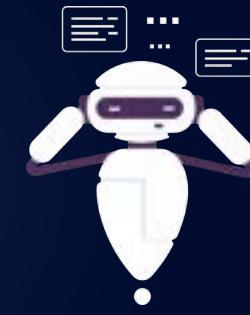


02

## TRANSFER LEARNING WITH DATA AUGMENTATION

- **VGG16 | ADAM LR: 0.001**
  - Fine Tuning
- **VGG16 | RMSprop LR: 1x10^-5**
  - Fine Tuning
- **RESNET 50 | ADAM LR: 0.001**
  - Fine Tuning
- **RESNET 50 | RMSprop LR: 1x10^-5**
  - Fine Tuning
- **INCEPTIONV3 | ADAM LR: 0.001**
  - Fine Tuning
- **INCEPTIONV3 | RMSprop LR: 1x10^-5**
  - Fine Tuning

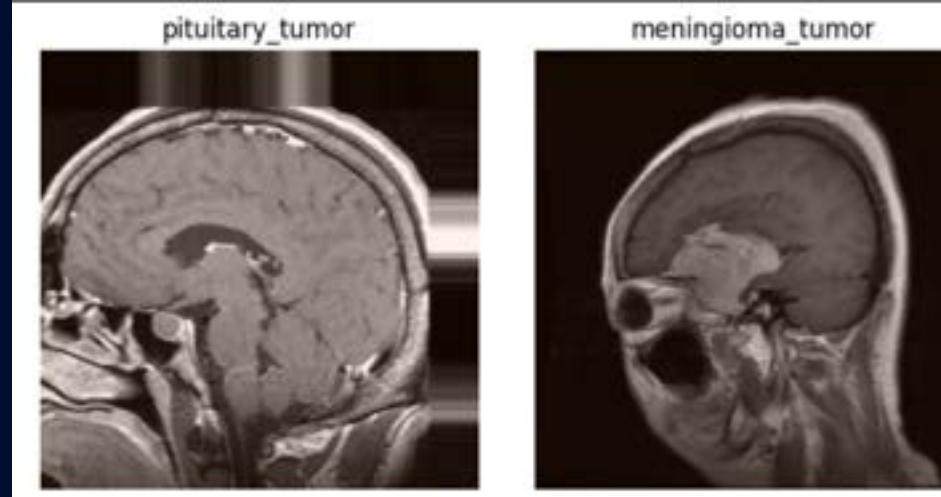
# DATA AUGMENTATION



## PARÁMETROS

- **Desplazamiento horizontal (width\_shift\_range):** Mueve la imagen horizontalmente dentro del rango especificado. Valor: 0.2
- **Desplazamiento vertical (height\_shift\_range):** Mueve la imagen verticalmente dentro del rango especificado. Valor: 0.2
- **Cizallamiento (shear\_range):** Aplica una transformación de cizallamiento a la imagen. valor: 0.2
- **Volteo horizontal (horizontal\_flip):** Voltea la imagen horizontalmente.
- **Relleno de bordes (fill\_mode):** Rellena los píxeles vacíos tras las transformaciones utilizando los valores más cercanos. Fill Mode: Nearest
- **Cambio de brillo (brightness\_range=[0.8, 2]):** Ajusta el brillo de la imagen dentro del rango especificado

```
# Combinar imágenes originales y aumentadas para el conjunto de entrenamiento  
X_train_combined = np.concatenate([X_train, augmented_train_images], axis=0)  
y_train_combined = np.concatenate([y_train, augmented_train_labels], axis=0)  
  
# Imprimir las formas finales de train y test  
print(f"X_train_combined shape: {X_train_combined.shape}")  
print(f"y_train_combined shape: {y_train_combined.shape}")  
  
X_train_combined shape: (5740, 224, 224, 3)  
y_train_combined shape: (5740,)
```



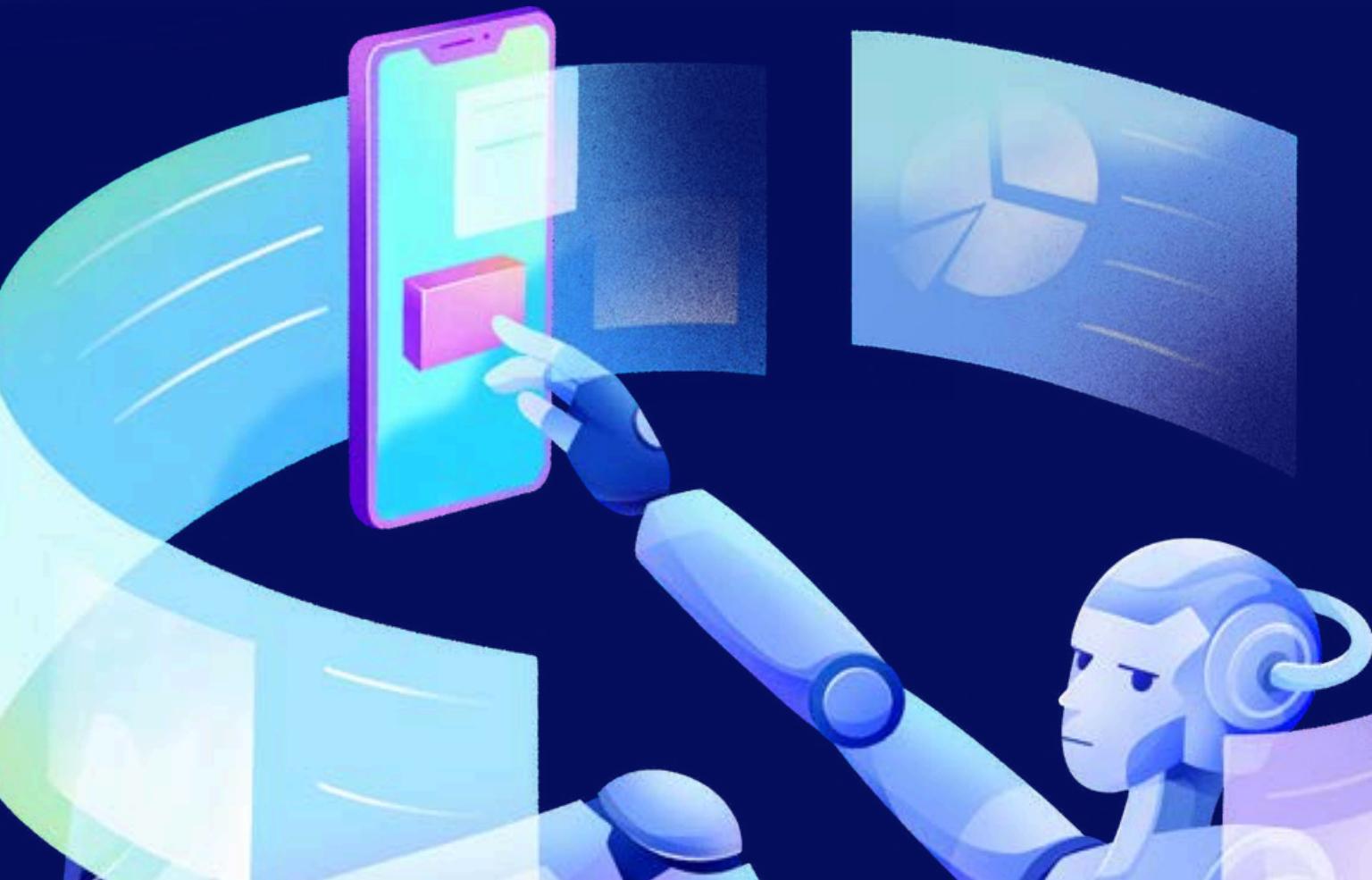
## DATOS GENERADOS

Datos originales  
**2870**

Offline Data aumentada  
**5740**

```
# Crear generador de datos  
data_generator = datagen.flow(X, y, batch_size=batch_size)  
  
if save_augmented:  
    ensure_directory_exists(augmented_save_dir)  
  
    steps_per_epoch = len(data_generator)  
    for i in range(steps_per_epoch):  
        batch_images, batch_labels = next(data_generator)  
        for j, image in enumerate(batch_images):  
            img = array_to_img(image)  
            img = img.resize(img_size)  
            category = categories[int(batch_labels[j])]  
            category_dir = os.path.join(augmented_save_dir, category)  
            ensure_directory_exists(category_dir)  
            img.save(os.path.join(category_dir, f"aug_{i}_{j}.png"))
```

# PERFORMANCE





# PERFORMANCE DE VGG16

**VGG16**  
**Complete Fine Tuning**  
**No Data Augmentation**  
**Adam lr = 0.001**

**VGG16**  
**Fine Tuning**  
**Data Augmentation**  
**Adam lr=0.001**

**VGG16**  
**Fine Tuning**  
**Data Augmentation**  
**RMSprop lr=0.,00001**

	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
GLIOMA	0,75	0,21	0,33	0,88	0,23	0,37	0,94	0,31	0,47
MENINGIOMA	0,69	0,93	0,80	0,68	0,89	0,77	0,79	0,96	0,86
NO TUMOR	0,62	0,94	0,75	0,64	0,93	0,76	0,73	0,99	0,84
PITUITARIO	0,91	0,65	0,76	0,80	0,70	0,75	0,85	0,89	0,87





# PERFORMANCE DE RESNET50

**RESNET 50**  
**Complete Fine Tuning**  
**No Data Augmentation**  
**Adam lr = 0.001**

**RESNET 50**  
**Fine Tuning**  
**Data Augmentation**  
**Adam lr=0.001**

**RESNET 50**  
**Fine Tuning**  
**Data Augmentation**  
**RMSprop lr=0.,00001**

	PRECISION	RECALL	FI-SCORE	PRECISION	RECALL	FI-SCORE	PRECISION	RECALL	FI-SCORE
GLIOMA	0,95	0,18	0,30	0,93	0,25	0,39	0,88	0,21	0,34
MENINGIOMA	0,65	0,90	0,75	0,74	0,95	0,83	0,75	0,95	0,84
NO TUMOR	0,61	0,96	0,75	0,68	0,98	0,80	0,68	0,99	0,81
PITUITARIO	0,96	0,62	0,75	0,90	0,82	0,86	0,89	0,85	0,87





# PERFORMANCE DE INCEPTION V3

**INCEPTION V3**  
Complete Fine Tuning  
No Data Augmentation  
Adam lr = 0.001

**INCEPTION V3**  
Fine Tuning  
Data Augmentation  
Adam lr=0.001

**INCEPTION V3**  
Fine Tuning  
Data Augmentation  
RMSprop lr=0.,00001

	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
GLIOMA	1,00	0,28	0,44	1,00	0,37	0,54	1,00	0,37	0,54
MENINGIOMA	0,71	0,96	0,82	0,72	0,98	0,83	0,76	0,94	0,84
NO TUMOR	0,74	0,98	0,82	0,76	1,00	0,86	0,70	0,99	0,82
PITUITARIO	0,89	0,86	0,88	1,00	0,81	0,90	0,88	0,78	0,83



# PERFORMANCE DE TODOS LOS MODELOS

	MACRO AVG PRECISION	WEIGHTED AVG PRECISION	MACRO AVG RECALL	WEIGHTED AVG RECALL	MACRO AVG F1-SCORE	WEIGHTED AVG F1-SCORE	ACCURACY
VGG16-I	0,74	0,73	0,68	0,70	0,66	0,66	0,70
VGG16-II	0,75	0,74	0,69	0,70	0,66	0,66	0,70
VGG16-III	0,82	0,82	0,79	0,79	0,76	0,76	0,79
RESNET 50-I	0,79	0,77	0,67	0,68	0,64	0,64	0,68
RESNET 50-II	0,81	0,80	0,75	0,76	0,72	0,72	0,76
RESNET 50-III	0,80	0,79	0,75	0,75	0,71	0,71	0,75
INCEPTION V3-I	0,83	0,83	0,77	0,77	0,74	0,74	0,77
INCEPTION V3-II	0,87	0,85	0,79	0,80	0,78	0,78	0,80
INCEPTION V3-III	0,83	0,83	0,77	0,78	0,76	0,76	0,78



DE LOS 9 MODELOS EVALUADOS PARA LA CLASIFICACIÓN DE TUMORES CEREBRALES SE ENCUENTRA QUE:

## ACCURACY GENERAL

Varía entre un 68% y 80%

## RENDIMIENTO GLOBAL

Los 3 mejores modelos se obtuvieron con Data Augmentation:

- Inception V3 (Adam- $\text{lr}:1\times10^{-3}$ ): **80% accuracy**
- VGG 16 (RMSprop- $\text{lr}:1\times10^{-5}$ ): **79% accuracy**
- Inception V3 (RMSprop- $\text{lr}:1\times10^{-5}$ ): **78% accuracy**

## DESEMPEÑO POR CLASE

La detección de gliomas mostró alta precisión pero bajo recall en todos los modelos, indicando que aunque las predicciones son precisas, se omiten muchas instancias verdaderas. En cambio, meningiomas y tumores pituitarios fueron clasificados con mayor éxito, alcanzando recalls de hasta 98% y altos f1-scores en varios modelos.

## RENDIMIENTO GLOBAL Y PÉRDIDAS EN TEST

- Inception V3 (Adam- $\text{lr}:1\times10^{-3}$ ): **80% accuracy - Test Loss: 3.43**
- VGG 16 (RMSprop- $\text{lr}:1\times10^{-5}$ ): **79% accuracy - Test Loss: 3.39**
- Inception V3 (RMSprop- $\text{lr}:1\times10^{-5}$ ): **78% accuracy - Test Loss: 1.24**



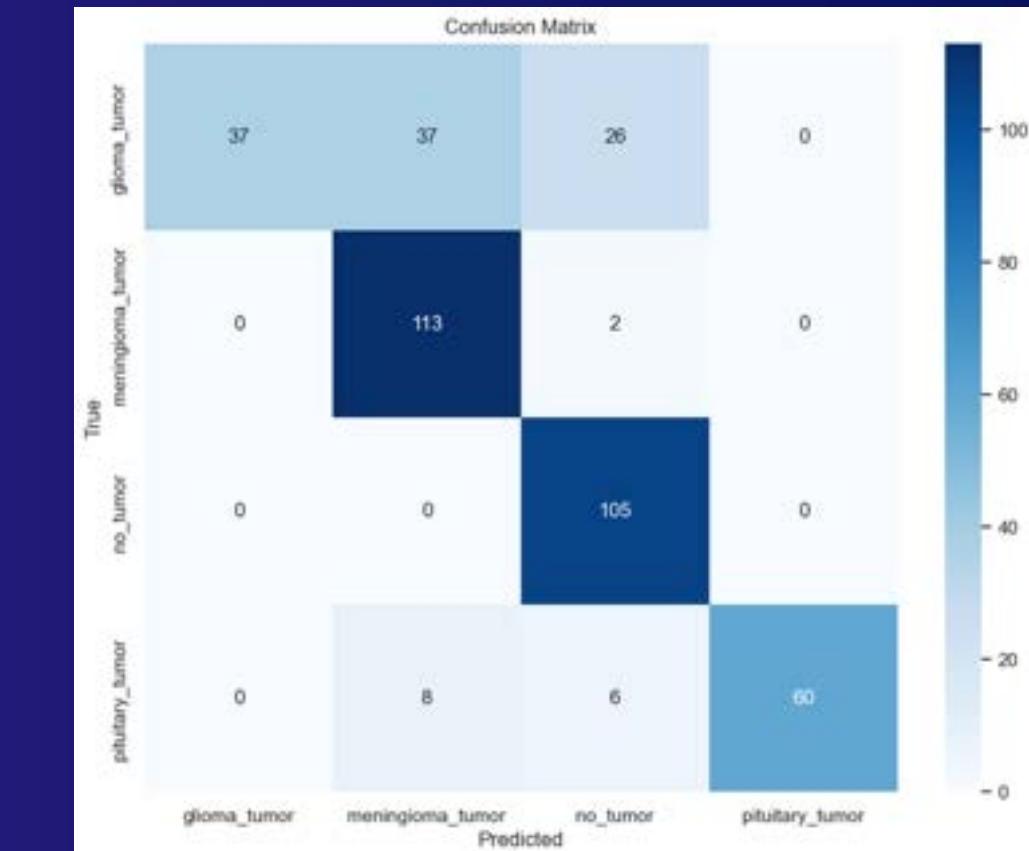
# RESULTADOS Y LOGROS

(01)

- Logramos obtener un modelo relativamente estable, que predice 78 % de las imágenes correctamente

(02)

- Se debe realizar un análisis más profundo de la clase glioma que es confundida con las demás en términos morfológicos del tipo de tumor, más específicamente con personas del área de neurología.





# CONCLUSIÓN

Concluimos que el mejor modelo es  
**Inception V3 (RMSPROP, LR: 1x10<sup>-5</sup>)**

- 78 % accuracy
- Test Loss: 1.24

A la hora de evaluar la calidad y consistencia de un modelo no solo debemos ver su métrica sino también su pérdida, ya que modelos con menores pérdidas en test, tienden a ser más confiables y estables en su desempeño, generalizando bien en datos no vistos en el entrenamiento y evitando el overfitting.

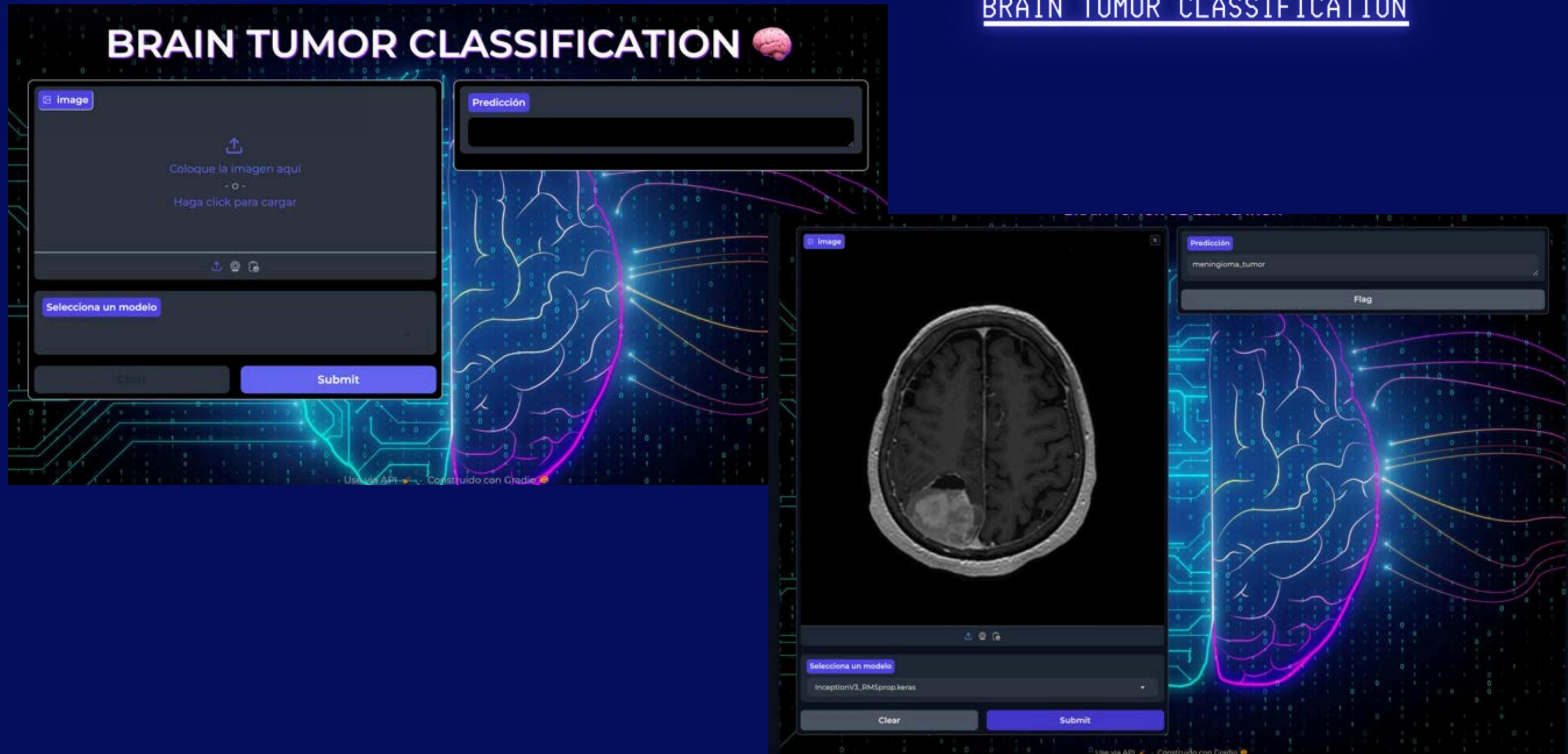


**GRACIAS!**



# DEMO

SE EXPUSO UNA API CON GRADIO Y USAMOS DE SERVER HUGGING FACE  
BRAIN TUMOR CLASSIFICATION



# VAMOS A LA DEMO

