

Inteligencia Artificial y Biomedicina: Evolución, actualidad y perspectivas futuras.

Nahuel Costa Cortez

Presentación

Nahuel Costa
Inteligencia Artificial y Biomedicina

2020 - 2023



Universidad de Oviedo



Doctorado en Inteligencia Artificial

2019 - 2020



Universidad de Oviedo

Máster en Ingeniería Informática

2015 - 2019



Universidad de Oviedo

Grado en Ingeniería Informática



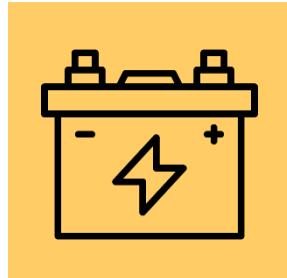
**Profesor en
Universidad de Oviedo**

Investigador en IA



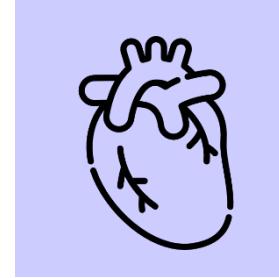
nahuelcosta.com

IA aplicada a monitorización de sistemas



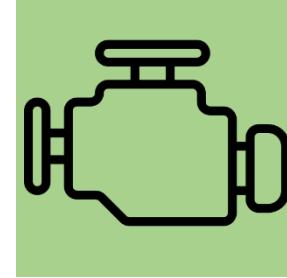
Baterías de Litio

Monitorización de datos
de celdas



Marcapasos

Monitorización de datos
intracardiacos



Motores de aviación

Monitorización de
sensores de aviones

Presentación

Nahuel Costa
Inteligencia Artificial y Biomedicina



Medtronic

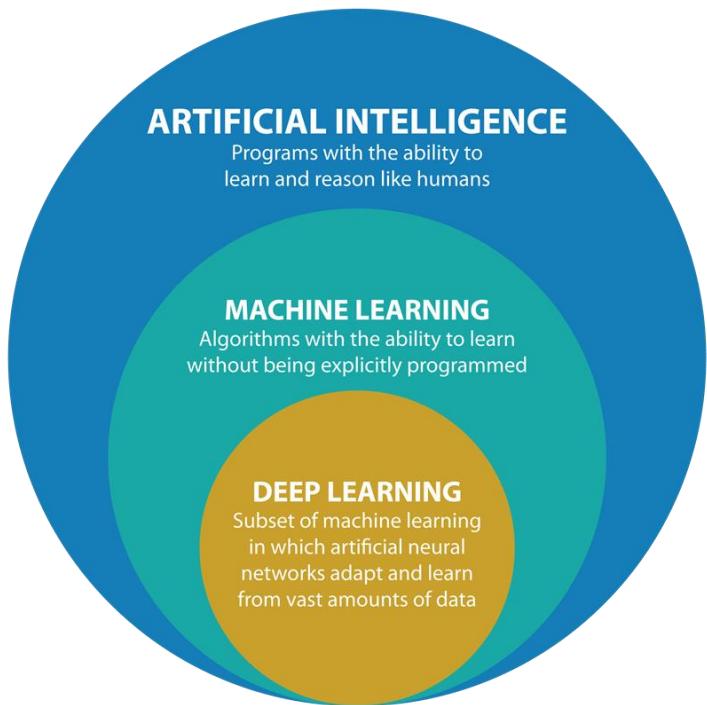


Inteligencia Artificial

Nahuel Costa
Inteligencia Artificial y Biomedicina

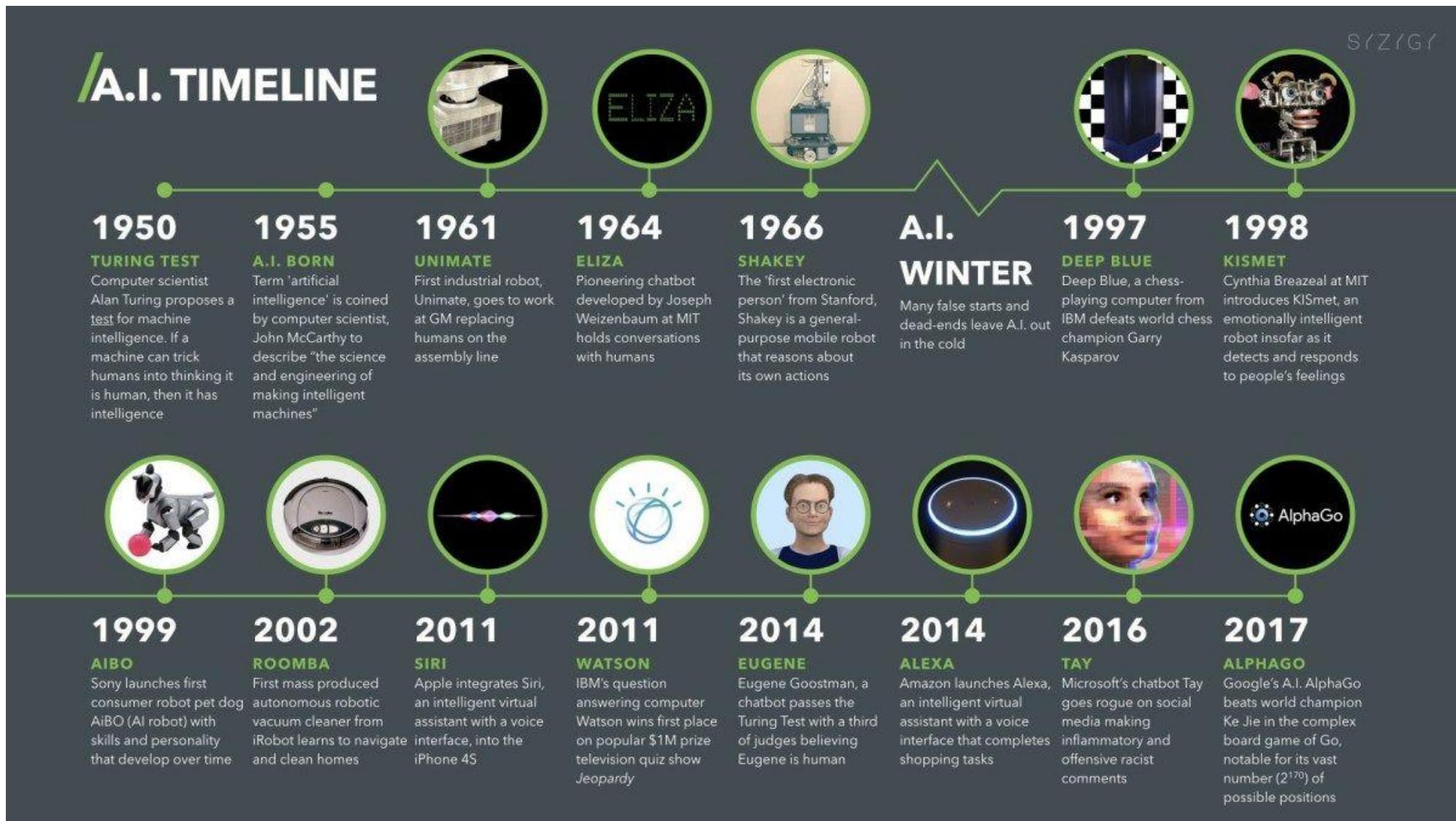


“Capacidad de las máquinas y los sistemas de software para realizar tareas que, en teoría, requerirían inteligencia humana para ser realizadas”



Inteligencia Artificial

Nahuel Costa
Inteligencia Artificial y Biomedicina



Evolución

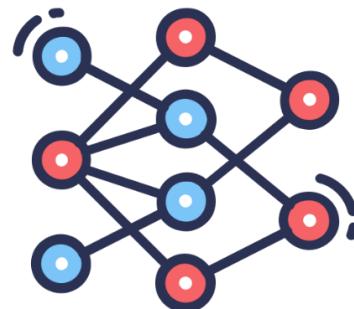
2012



IMAGENET

Reconocimiento de objetos
Desbloqueo móvil
Sistemas de recomendación

2015



EVOLUCIÓN MODELOS

Autocorrector móvil
Asistentes virtuales
Detección de spam

2016



MODELOS GENERATIVOS

Filtros en redes sociales
Aplicaciones de diseño

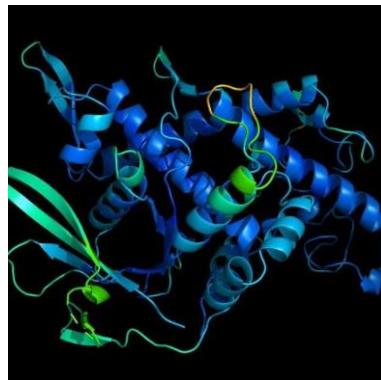
Evolución

2017



Transformers

AlphaFold



Plegado de proteínas

ChatGPT



Predicción de texto

Github Copilot



Predicción de código

Dalle2, Midjourney...

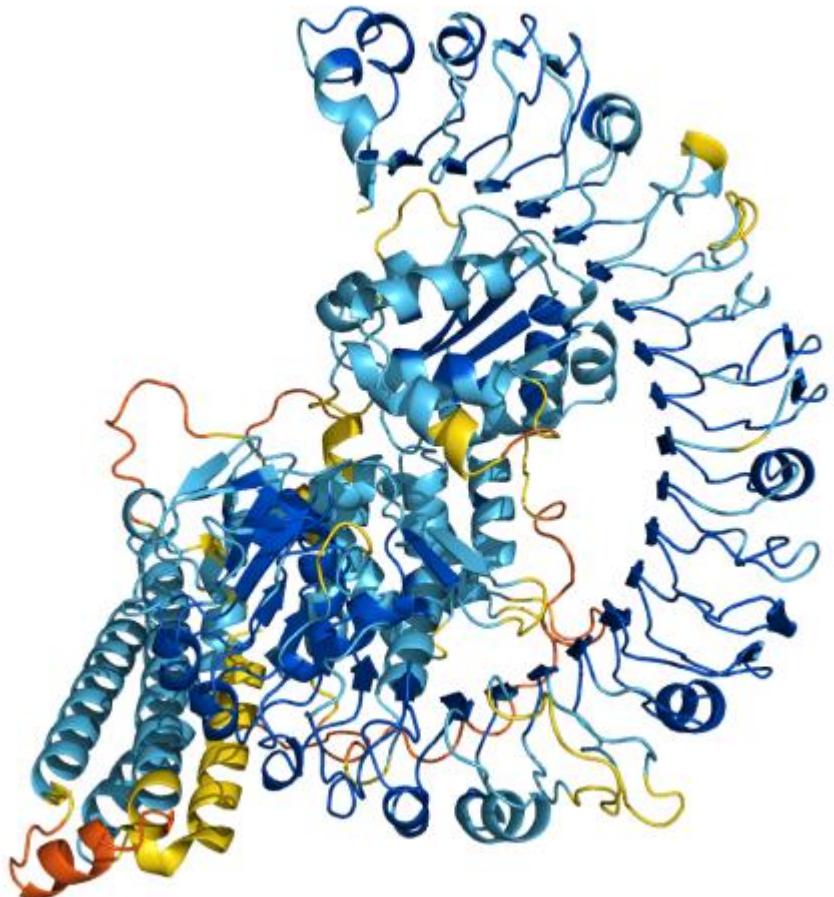


Generación de imágenes

AlphaFold



DeepMind



PLEGADO DE PROTEÍNAS

Proteínas - Secuencias de aminoácidos



PLEGADO DE PROTEÍNAS

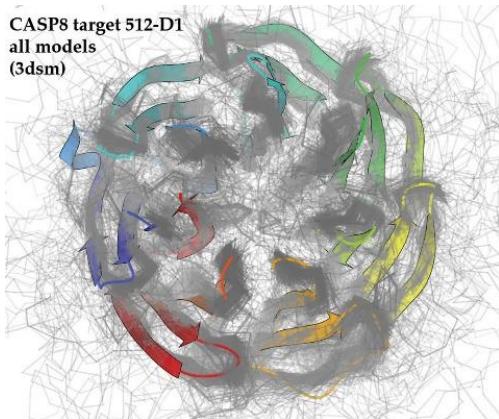
Proteínas - Secuencias de aminoácidos



Mal plegado

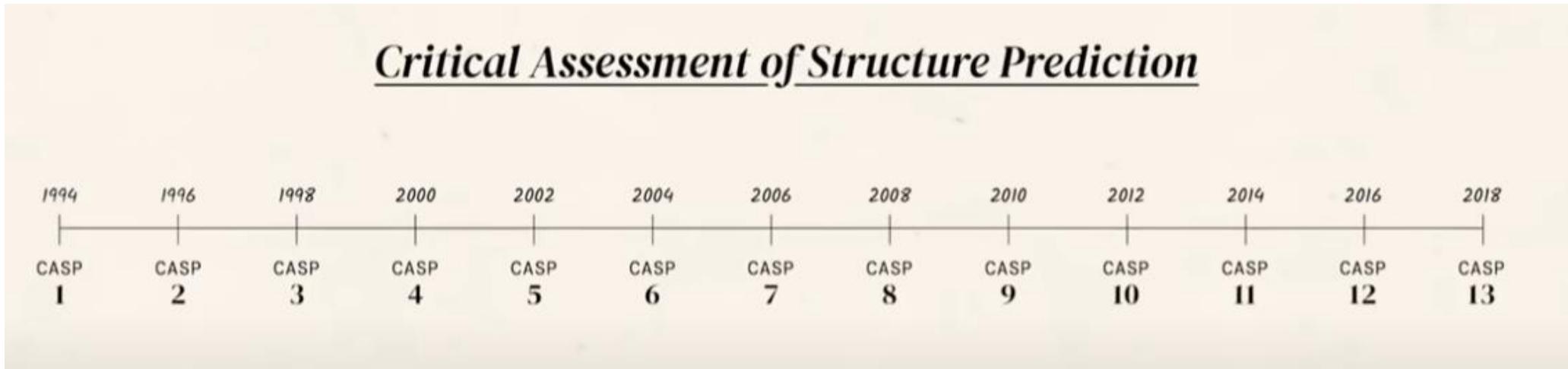


Enfermedades: Alzheimer, Parkinson...



COMPETICIÓN CASP

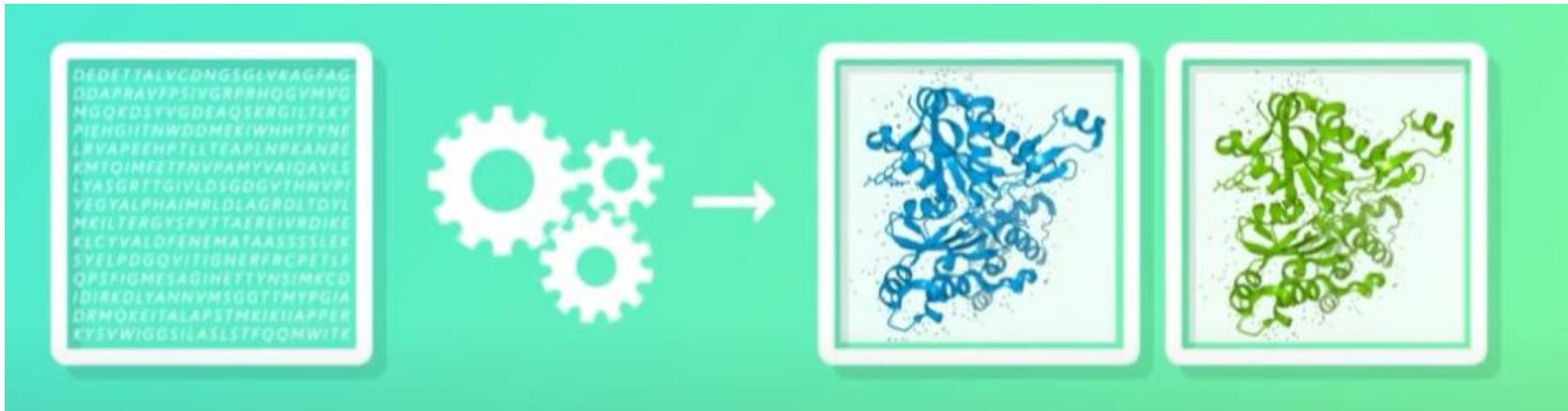
Métodos tradicionales -
Computacionalmente muy costosos



COMPETICIÓN CASP

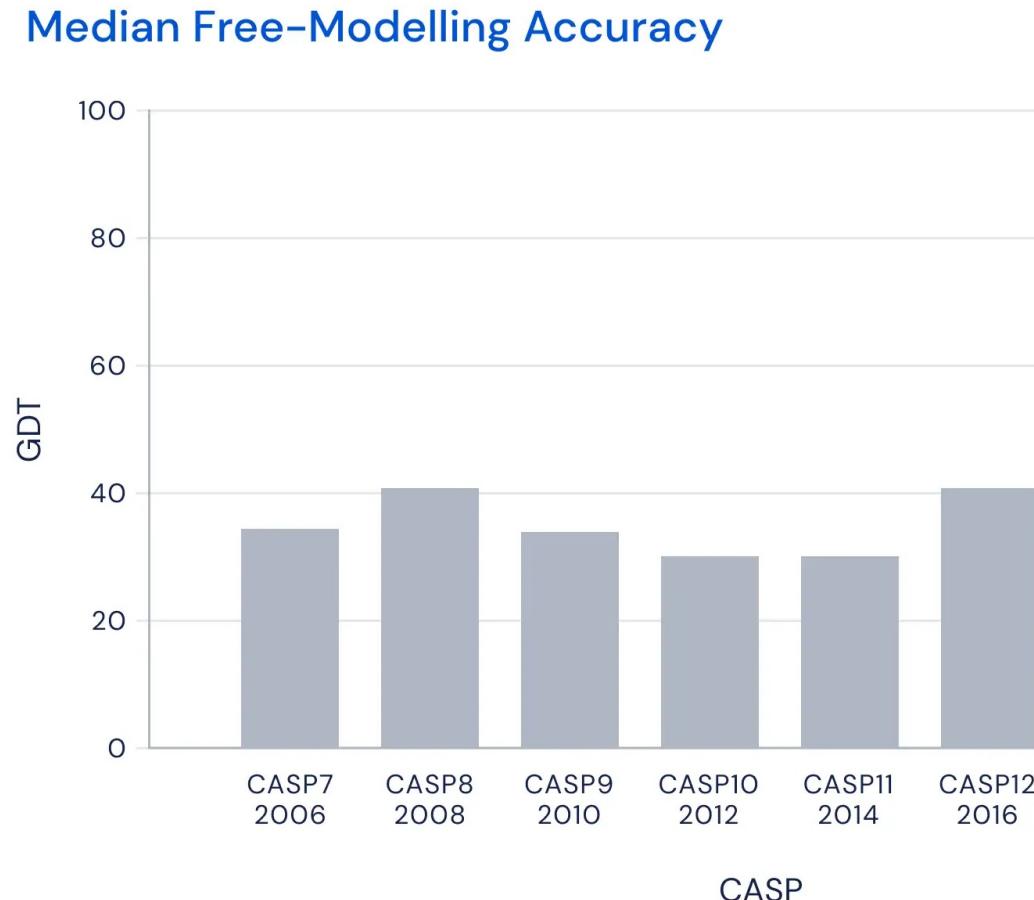
SECUENCIA DE AMINOÁCIDOS

ESTRUCTURA 3D PROTEINA



Métrica - Global Distance Test (GDT)

Indica el porcentaje de aminoácidos que se localizan en la posición correcta

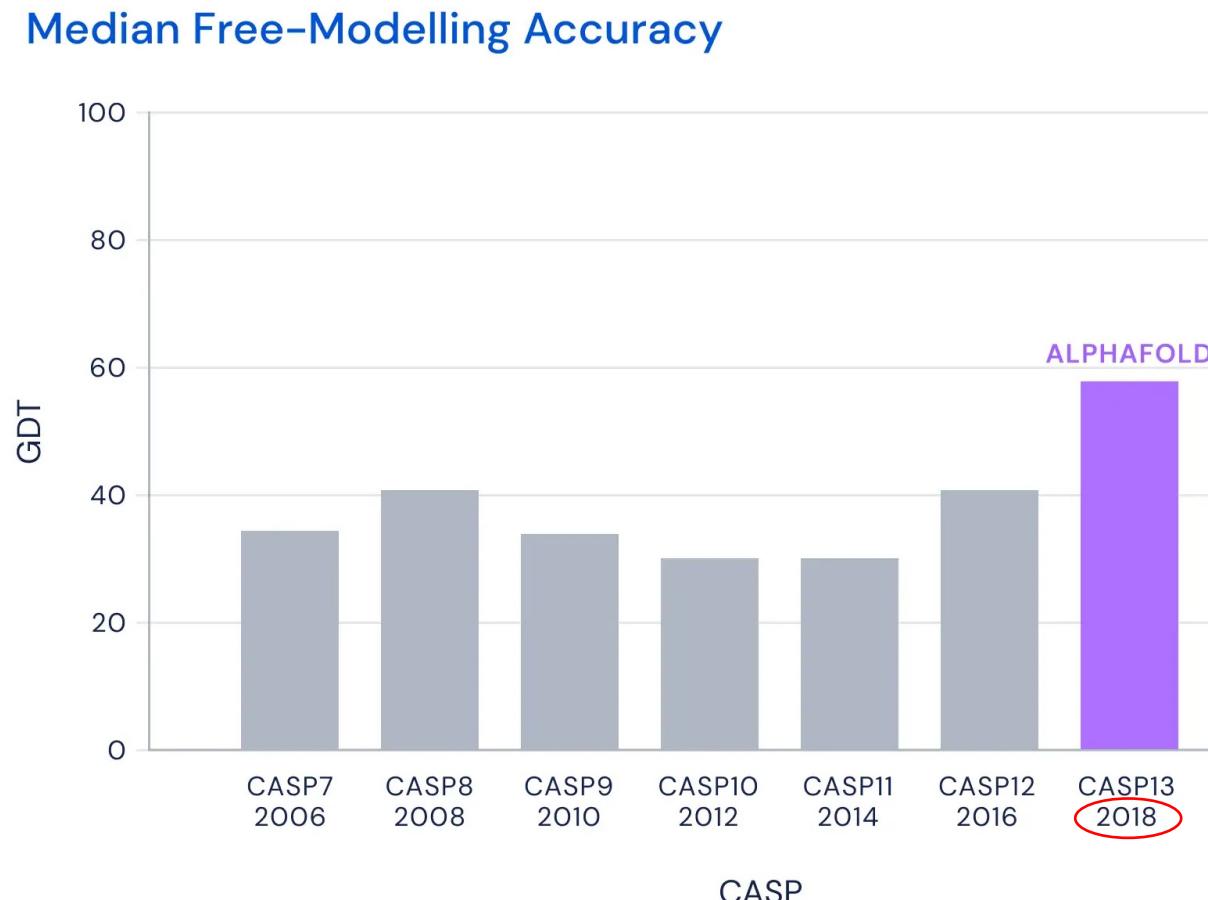


Métrica - Global Distance Test (GDT)

Indica el porcentaje de aminoácidos que se localizan en la posición correcta



DeepMind

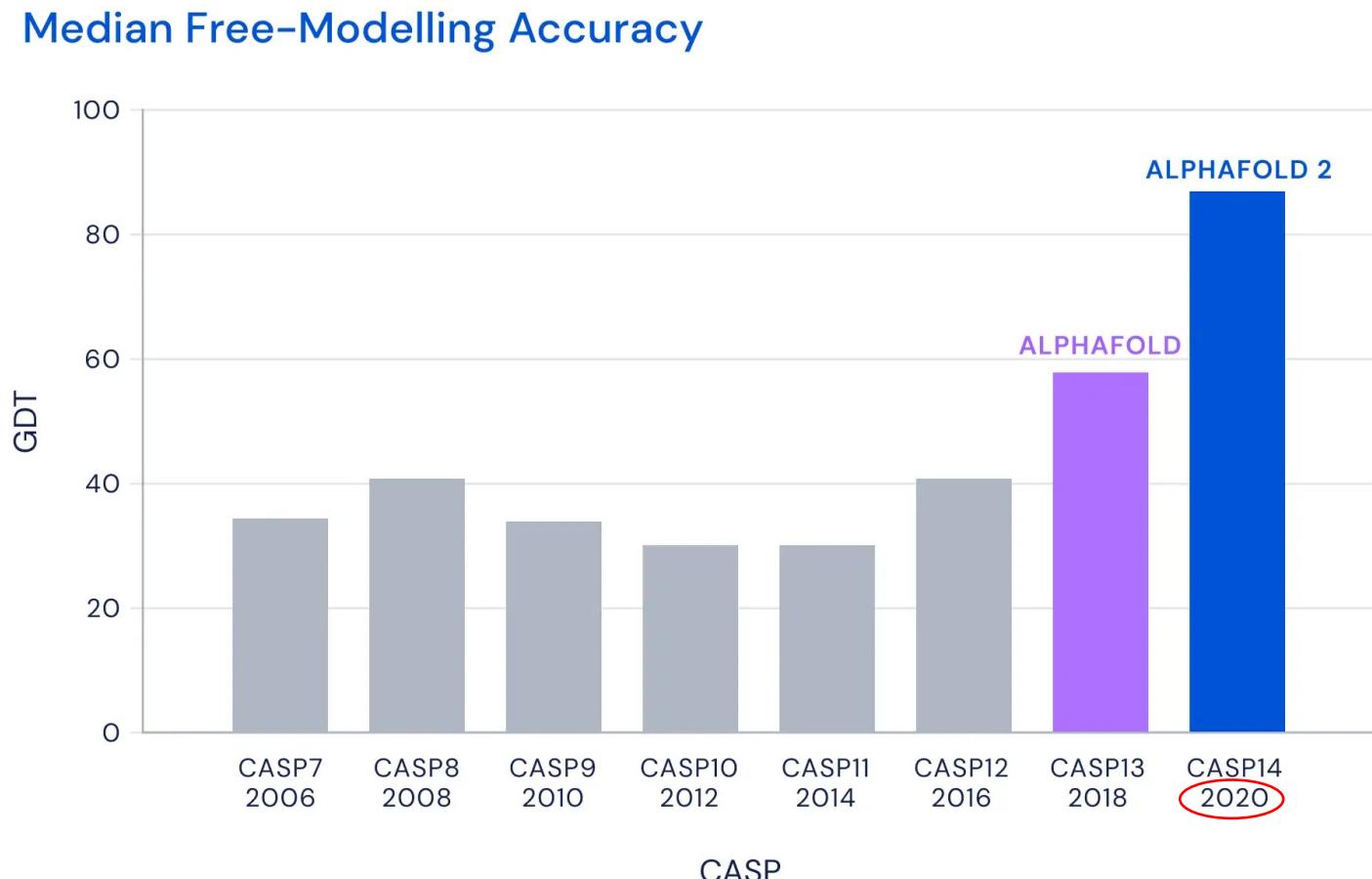


Métrica - Global Distance Test (GDT)

Indica el porcentaje de aminoácidos que se localizan en la posición correcta



¡Problema resuelto!



ALPHAFOLD 1

Convertir una secuencia unidimensional a
una estructura tridimensional

ALPHAFOLD 1

Convertir una secuencia unidimensional a
una estructura tridimensional



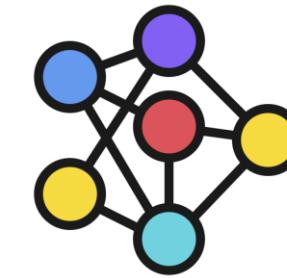
Convertir una imagen a otra imagen

AlphaFold

Nahuel Costa
Inteligencia Artificial y Biomedicina

Tomar una imagen y modificar su estilo

Content image



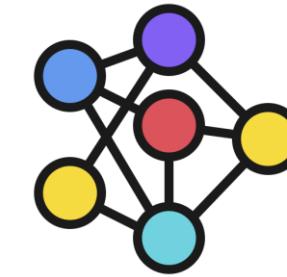
Output image



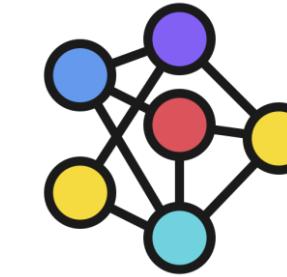
AlphaFold

Nahuel Costa
Inteligencia Artificial y Biomedicina

Tomar una imagen y modificar su estilo



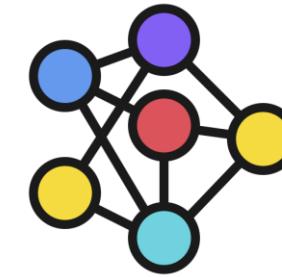
Tomar una imagen y generar un mapa de segmentación



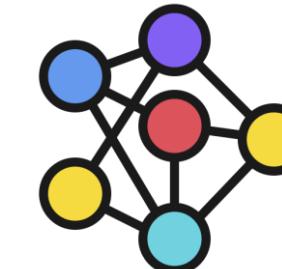
AlphaFold

Nahuel Costa
Inteligencia Artificial y Biomedicina

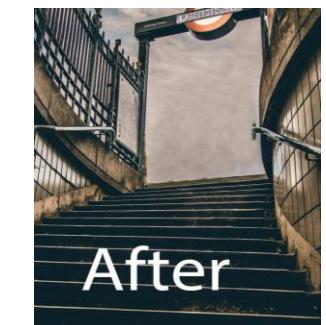
Tomar una imagen y modificar su estilo



Tomar una imagen y generar un mapa de segmentación

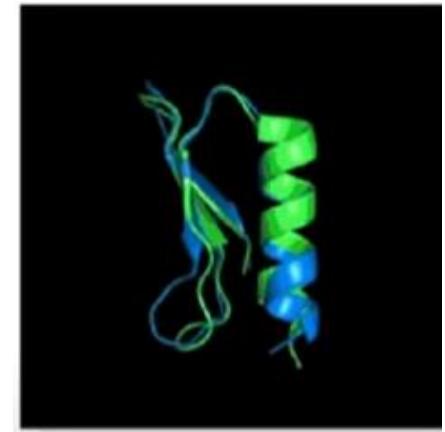
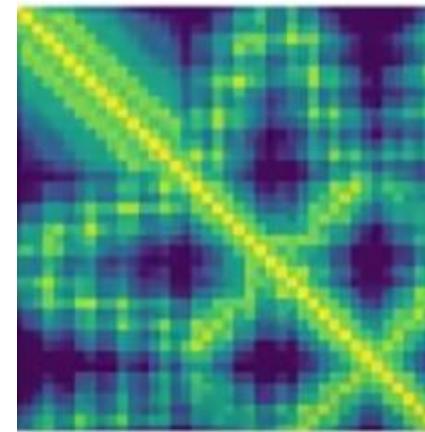
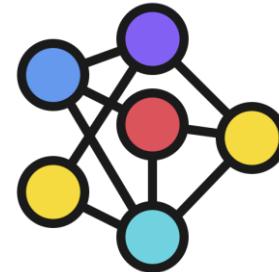
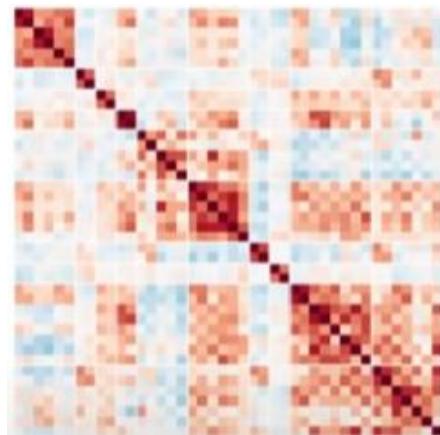


Tomar una imagen y borrar partes

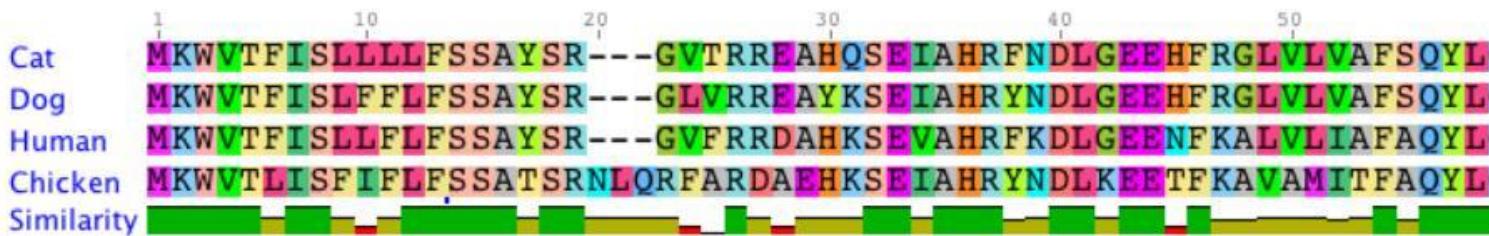
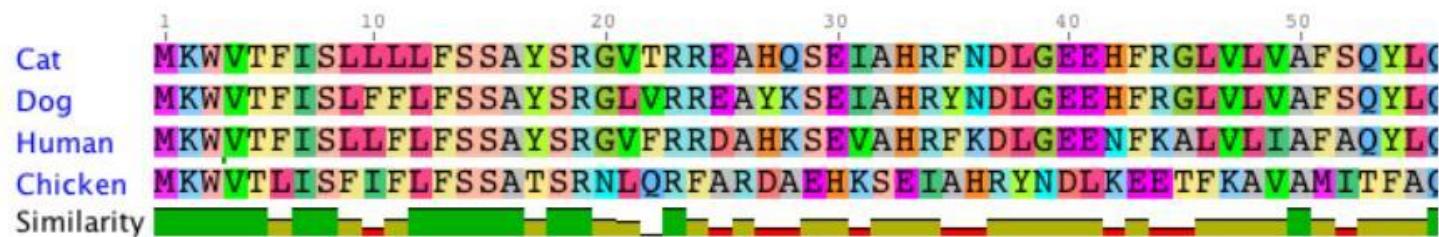


ALPHAFOLD 1

SQETRKKCTEMKKKFKN
CEVRCDESNCVEVRCS
DTKYTLC



Multiple sequence alignment



Multiple Sequence Alignment (MSA)



Aminoácidos que coevolucionan

Qué interacción se produce dentro de la estructura tridimensional

Predecir plegado

Multiple Sequence Alignment (MSA)

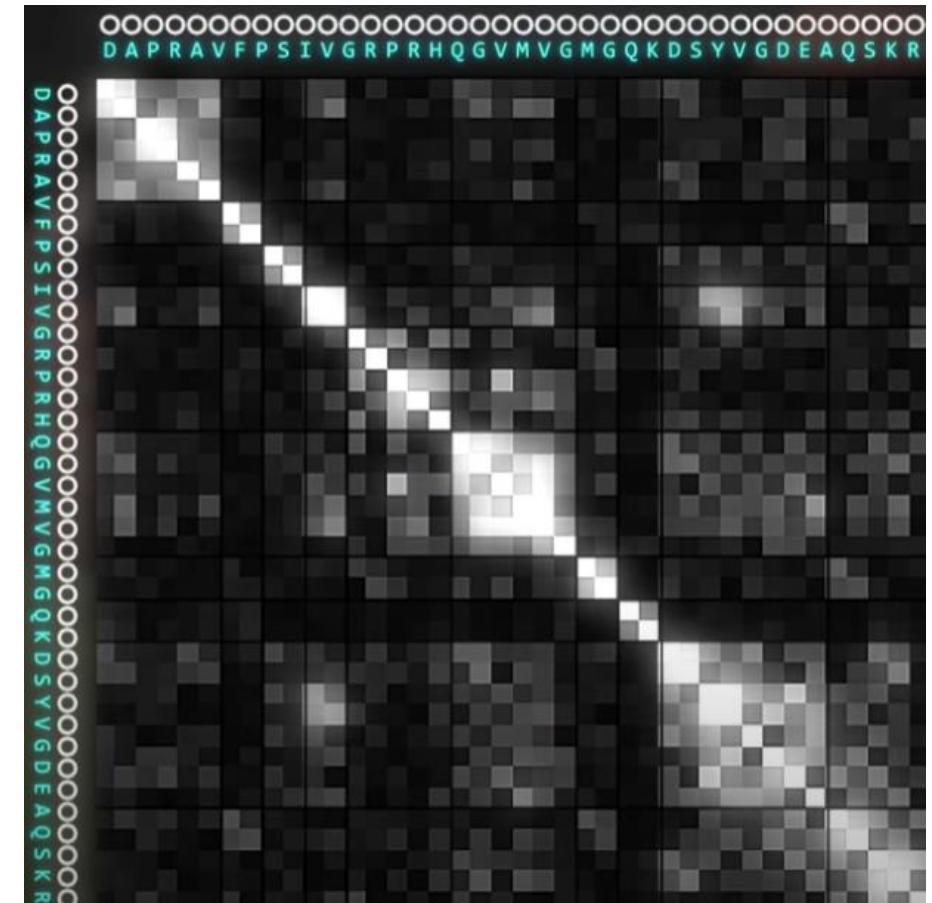


Aminoácidos que coevolucionan

Qué interacción se produce dentro de la estructura tridimensional

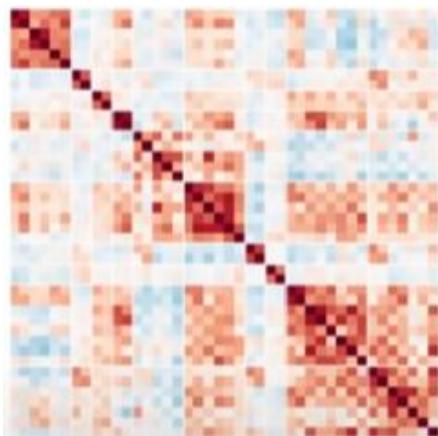
Predecir plegado

Interacción de aminoácidos

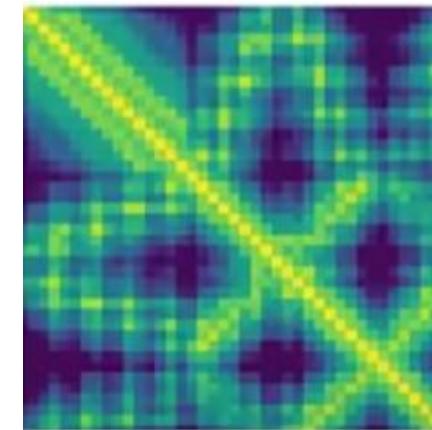
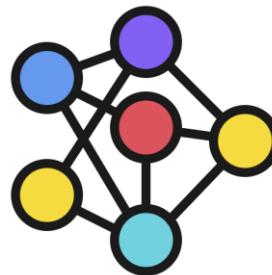


AlphaFold

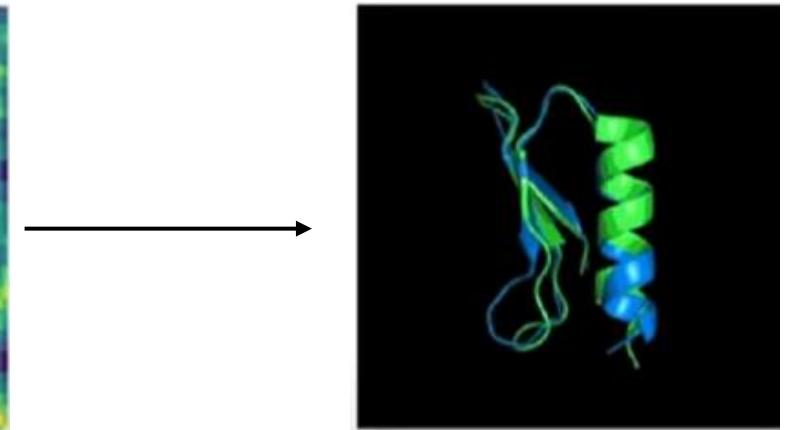
Nahuel Costa
Inteligencia Artificial y Biomedicina



INPUT
Información de la secuencia
de aminoácidos



OUTPUT
Matriz de distancia de la
proteína



OUTPUT FINAL
Estructura 3D de
la proteína

ALPHAFOLD 2

Convertir una secuencia unidimensional a
una estructura tridimensional



Modelarlo como un problema de texto



Transformers

ALPHAFOOLD 2

Convertir una secuencia unidimensional a
una estructura tridimensional



Modelarlo como un problema de texto



Transformers



Permiten atender a las partes más
importantes de una secuencia

ALPHAFOOLD 2

Convertir una secuencia unidimensional a
una estructura tridimensional



Modelarlo como un problema de texto



Transformers



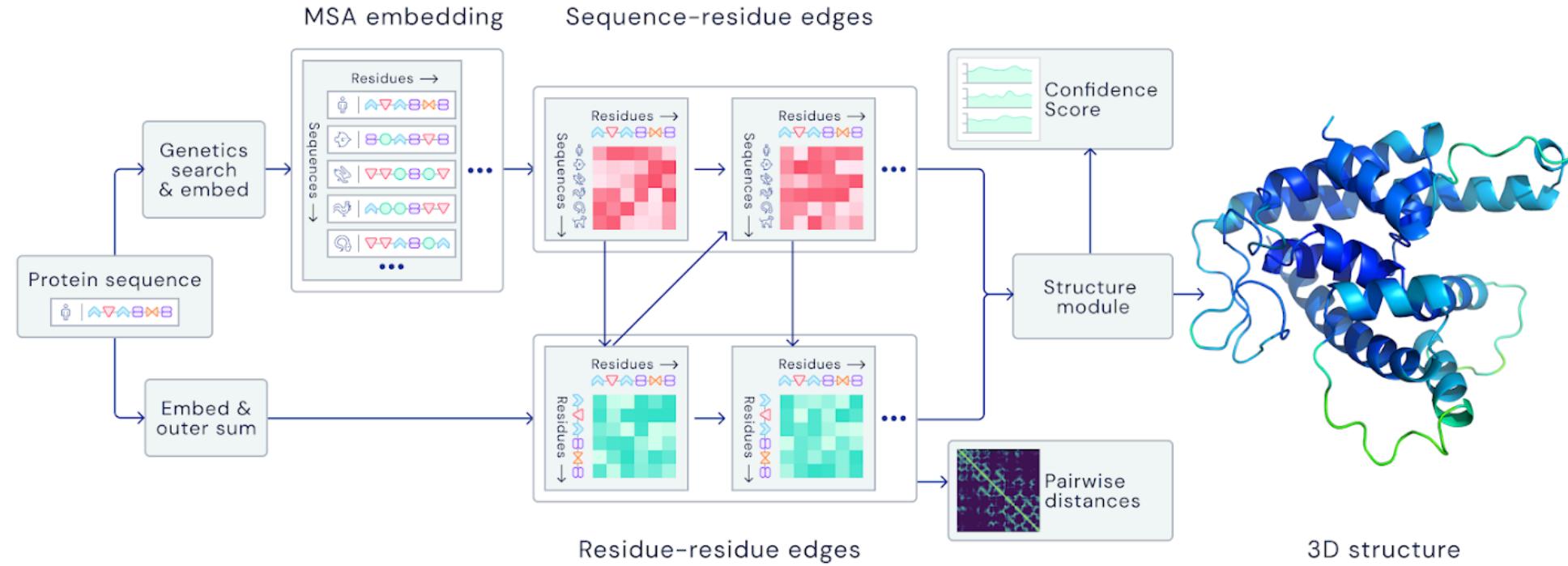
Permiten atender a las partes más
importantes de una secuencia



Pueden ser entrenados en paralelo

AlphaFold

Nahuel Costa
Inteligencia Artificial y Biomedicina



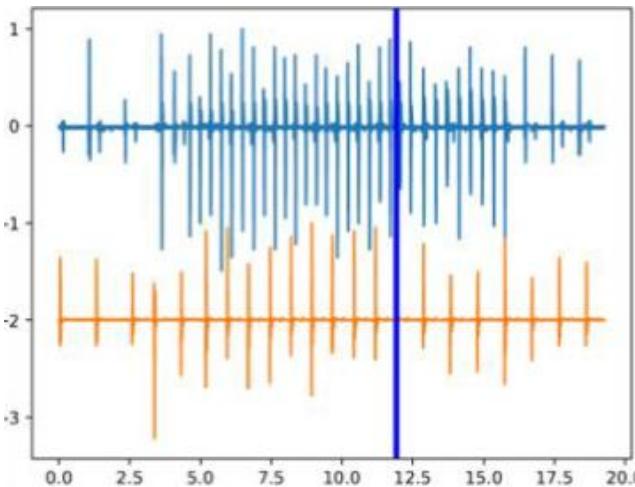
Encuentra secuencias similares a la entrada y pasa esa información a otra red neuronal que produce la estructura 3D final

A detailed anatomical illustration of the human torso from the neck to the mid-abdomen. The heart is prominently displayed in the center, colored in shades of red and pink. The lungs are visible on either side of the heart, appearing as large, translucent blue and purple structures. The ribcage and spine are shown as a network of white lines against a dark blue background.

Fibrilación Auricular

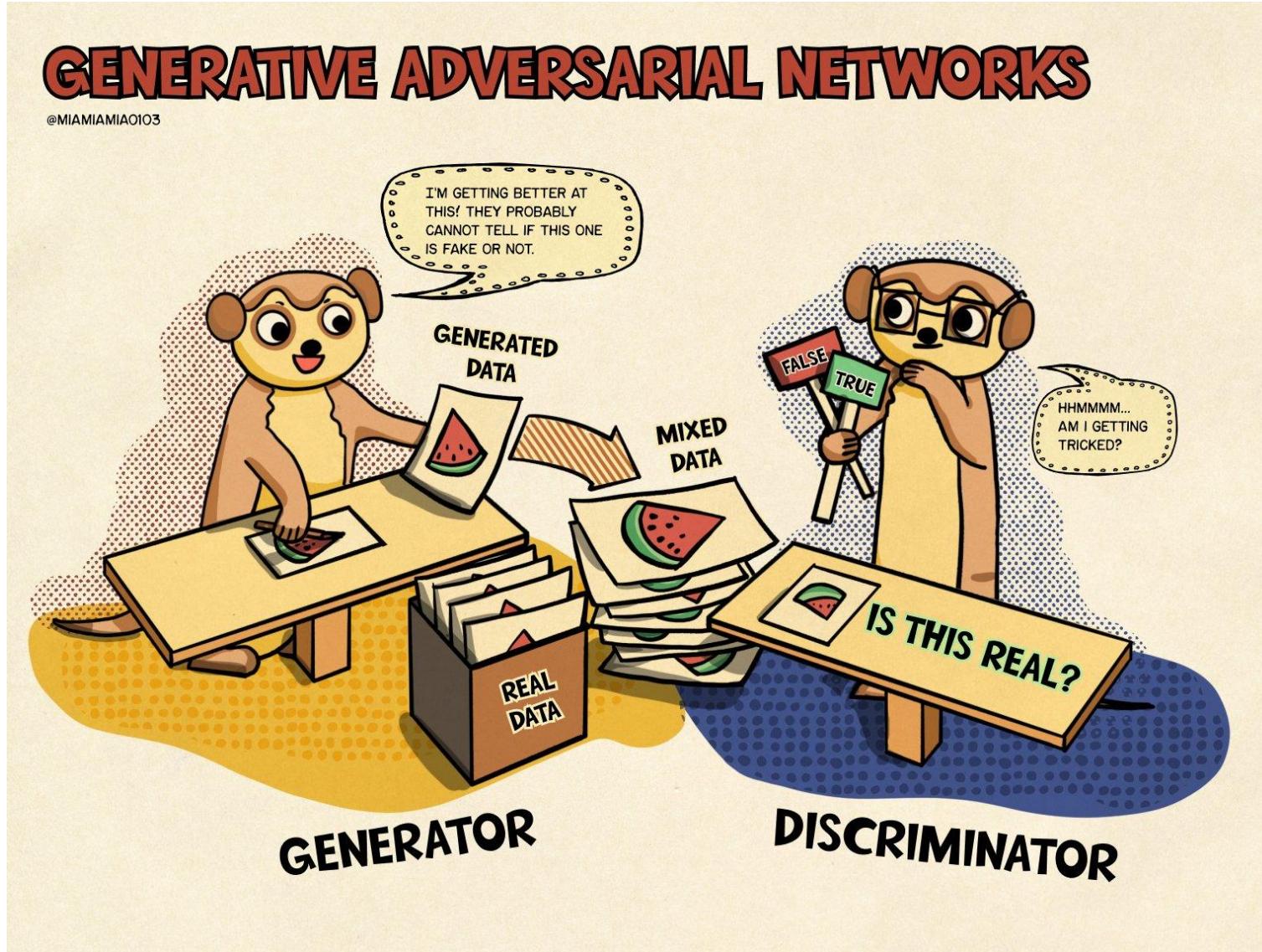
Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina

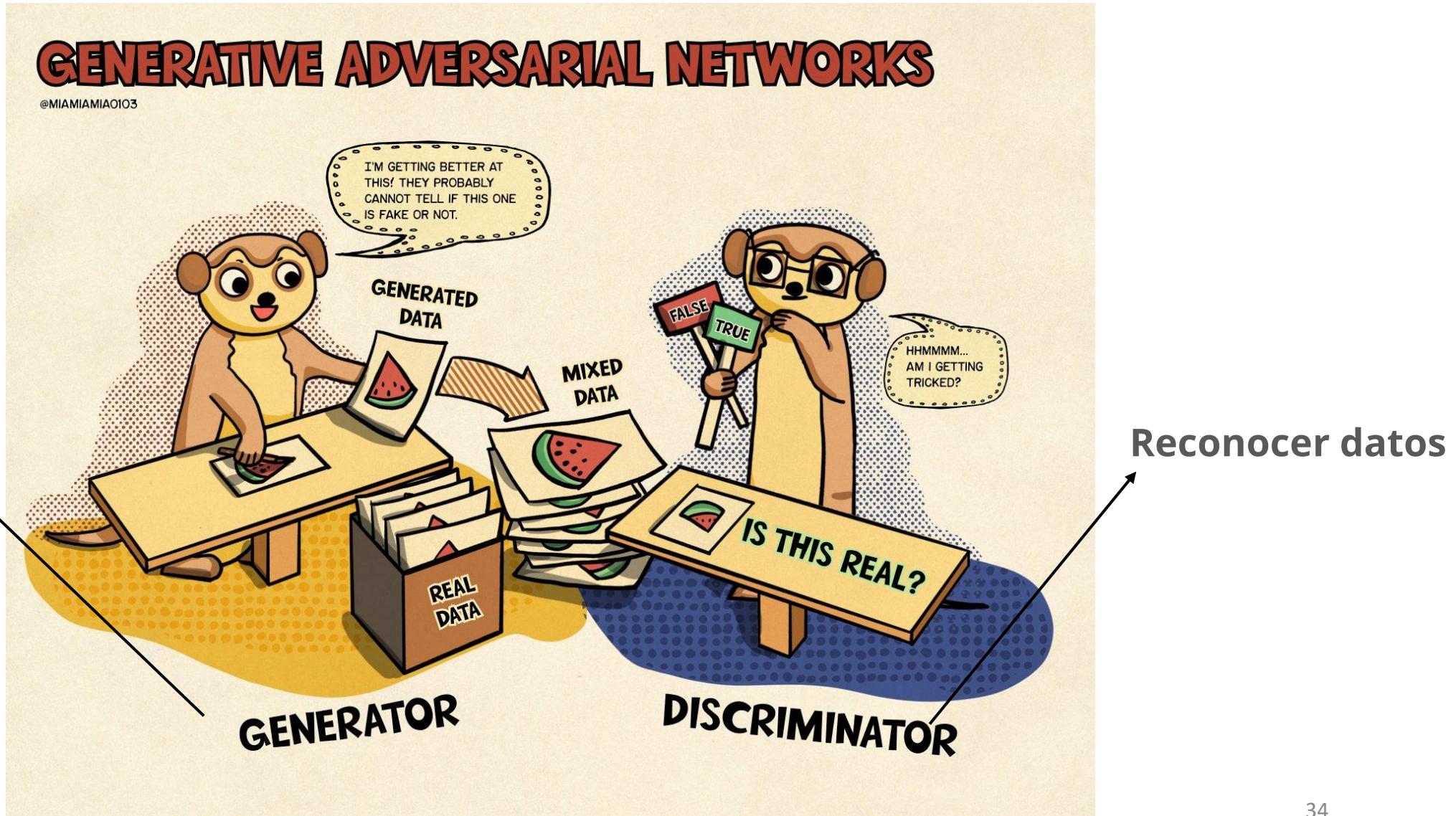


DATOS INCOMPLETOS
REGISTROS LIMITADOS

GANs

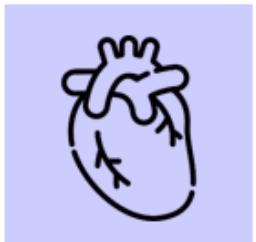


GANs



Fibrilación Auricular

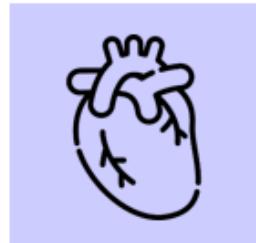
Nahuel Costa
Inteligencia Artificial y Biomedicina



Atrial Fibrillation

Class 1

Class 2



Class 3

Class 4

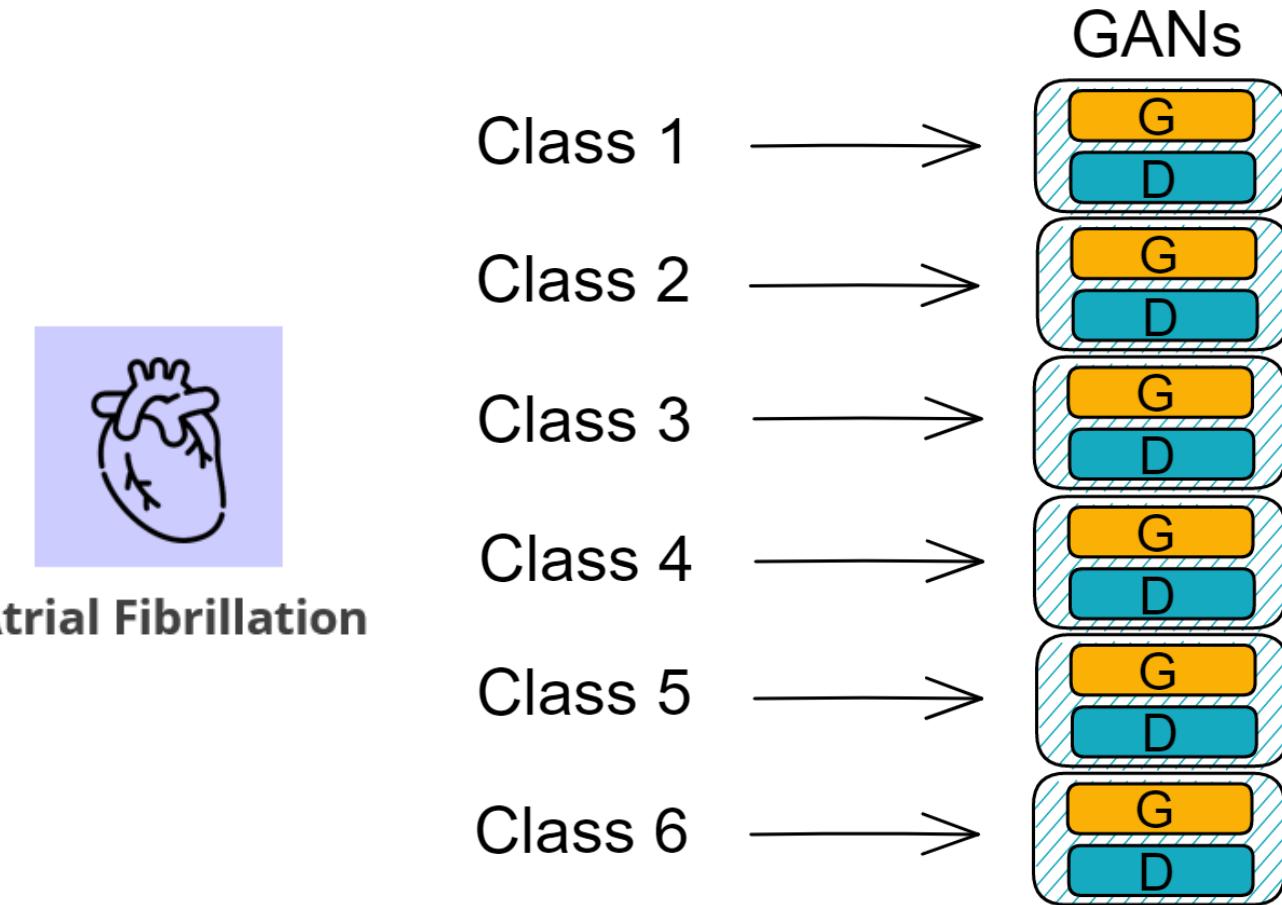
Class 5

Class 6

Atrial Fibrillation

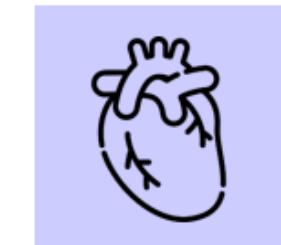
Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina

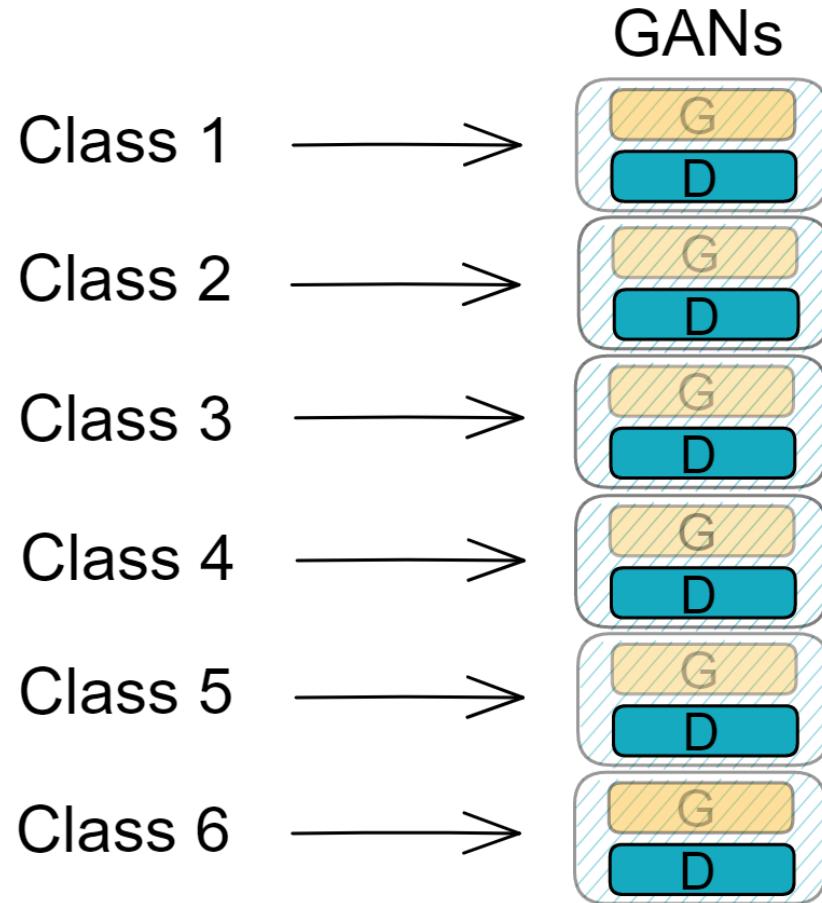


Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina

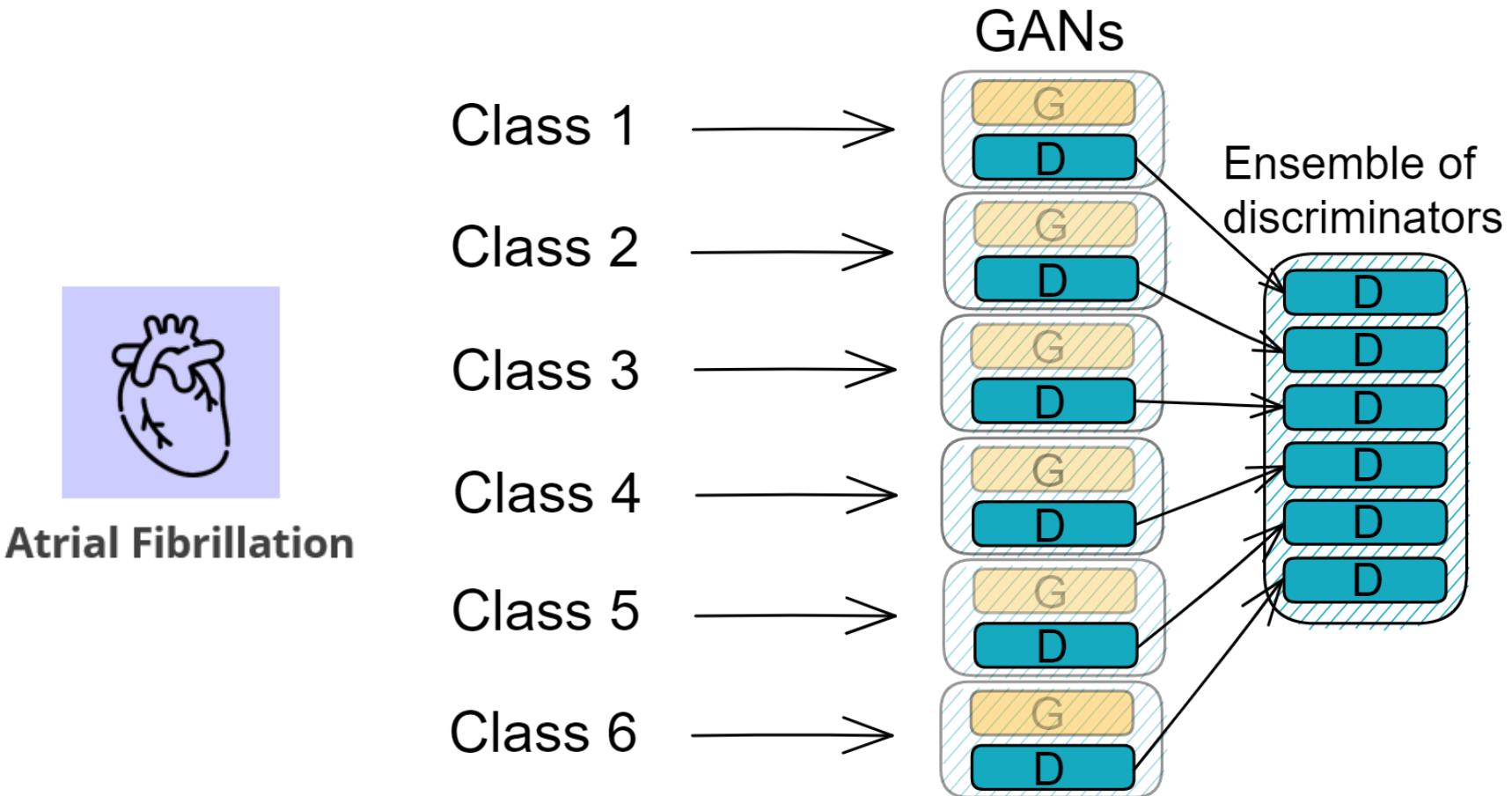


Atrial Fibrillation



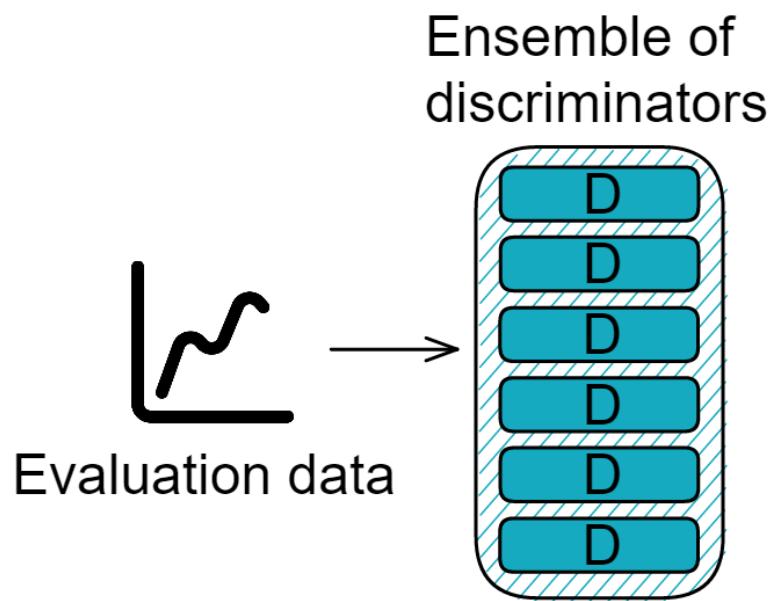
Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina



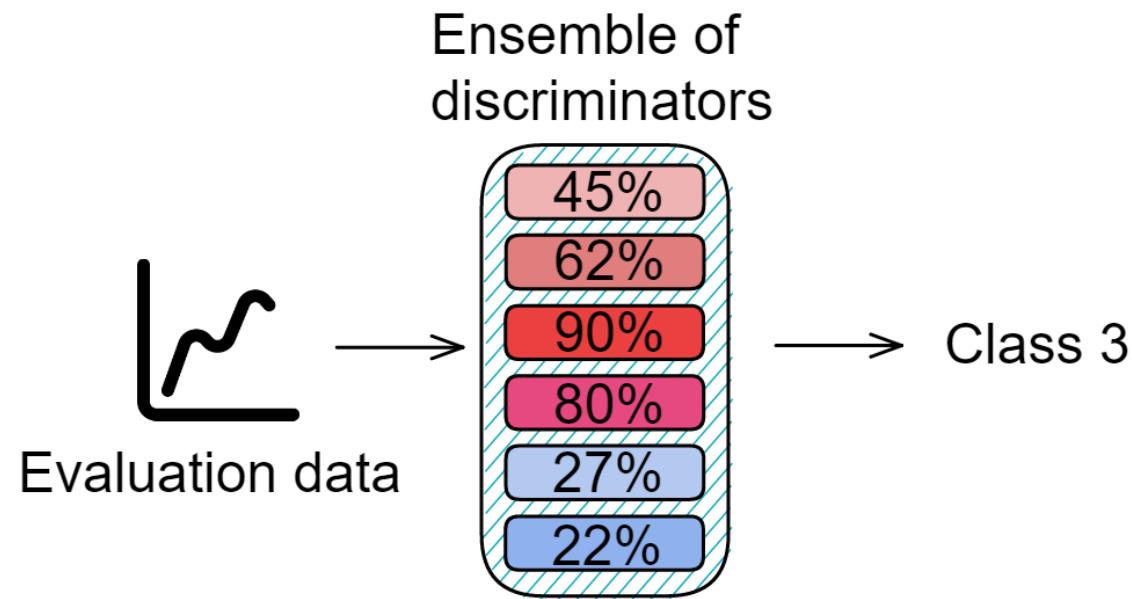
Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina



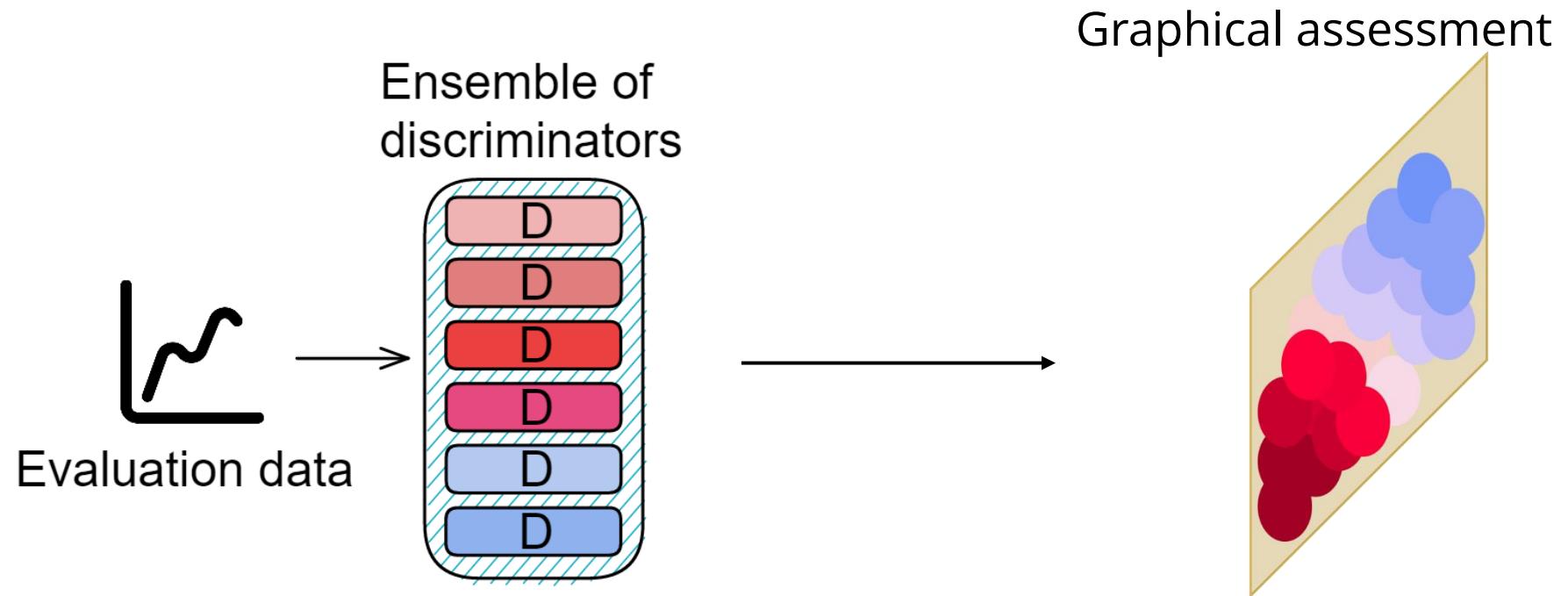
Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina



Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina



Class 1

Class 2

Class 3

Class 4

Class 5

Class 6



Atrial Fibrillation

α = velocidad de progresión hacia arritmia permanente

β = tiempo medio entre episodios de arritmia

Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina



Atrial Fibrillation

Class 1

Class 2

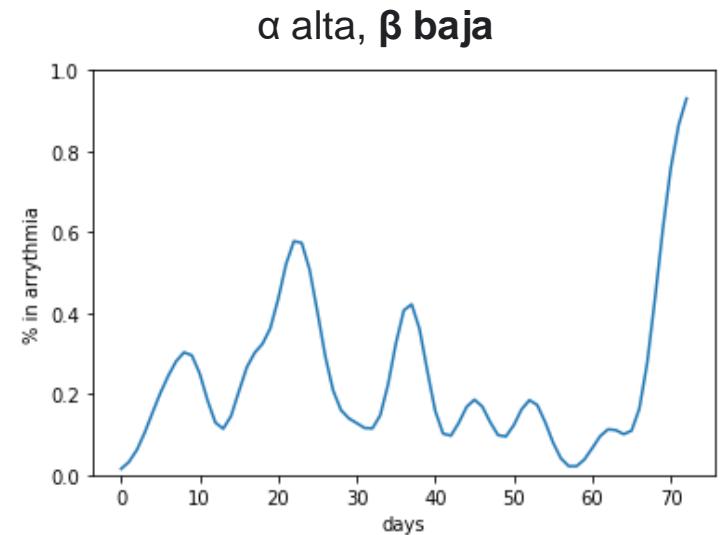
Class 3

Class 4

Class 5

Class 6

Muchos episodios,
progresión rápida
hacia permanente



α = velocidad de progresión hacia arritmia permanente

β = tiempo medio entre episodios de arritmia

Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina



Atrial Fibrillation

Class 1

Muchos episodios,
progresión rápida
hacia permanente

Class 2

Class 3

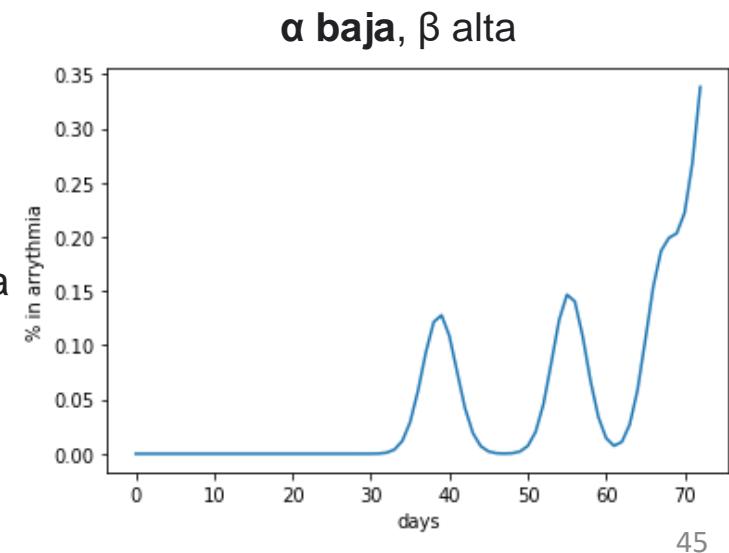
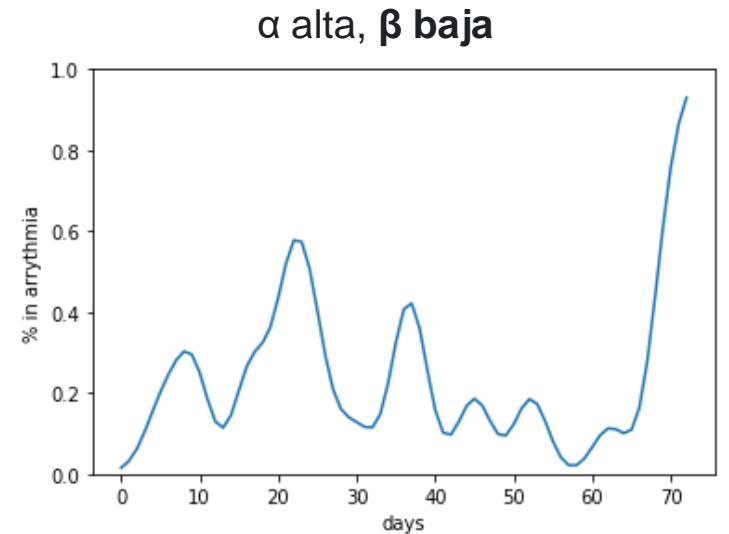
Class 4

Class 5

Class 6

α = velocidad de progresión hacia arritmia permanente

β = tiempo medio entre episodios de arritmia



Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina



Atrial Fibrillation

α = velocidad de progresión hacia arritmia permanente

β = tiempo medio entre episodios de arritmia

Class 1

Muchos episodios,
progresión rápida
hacia permanente

Class 2

Class 3

Criticidad es la misma
Independientemente del
número de episodios



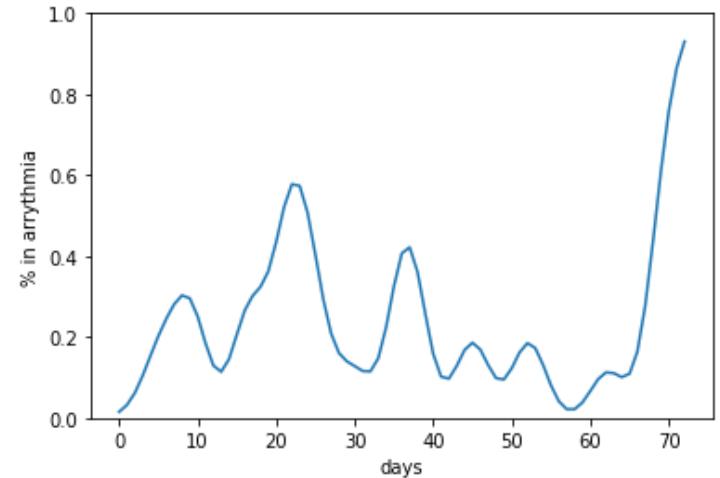
Class 4

Class 5

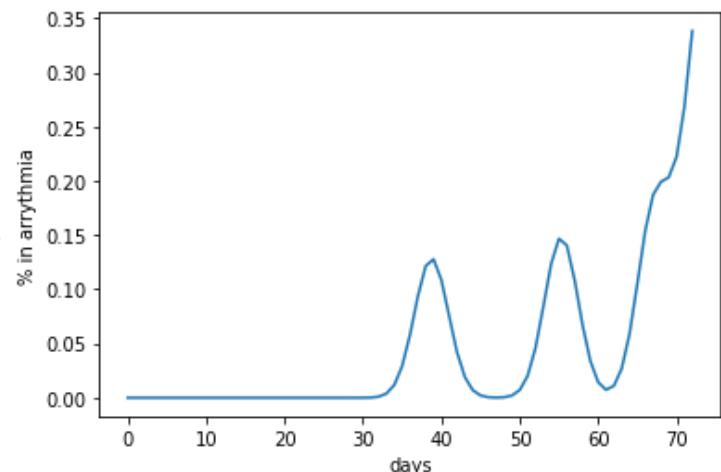
Class 6

Pocos episodios,
pero también
progresión rápida hacia
permanente

α alta, β baja



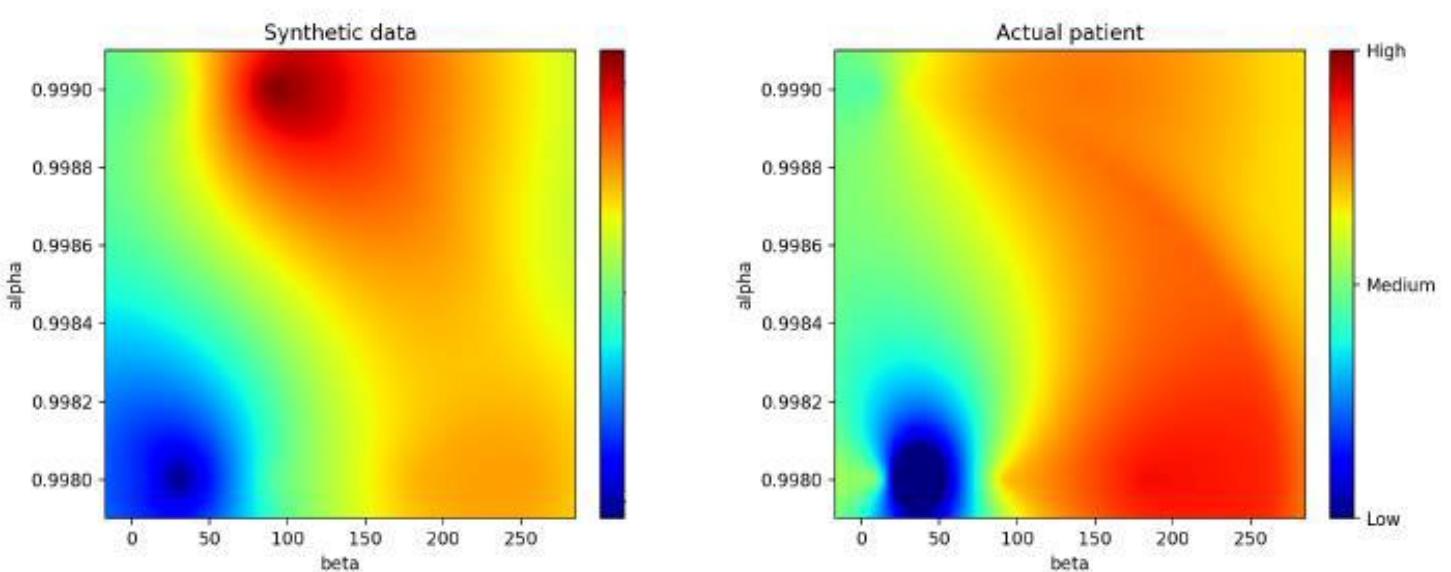
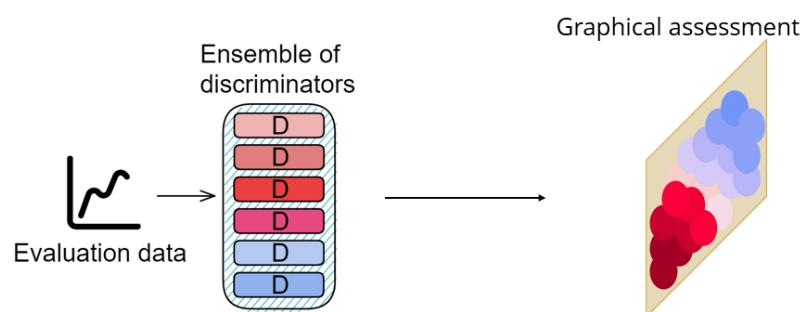
α baja, β alta



Fibrilación Auricular

Nahuel Costa
Inteligencia Artificial y Biomedicina

Mapa gráfico de las
activaciones de las últimas
capas de cada discriminador

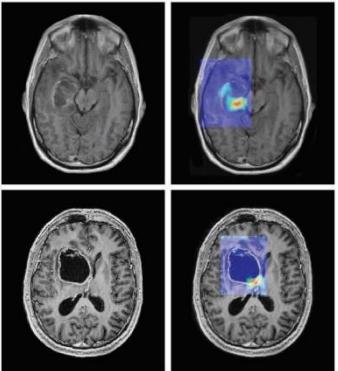




Otras aplicaciones

Otras aplicaciones

Nahuel Costa
Inteligencia Artificial y Biomedicina



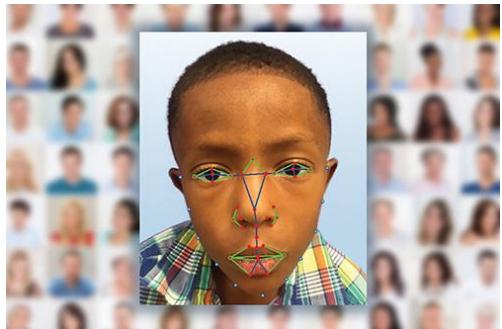
Análisis de imágenes médicas



Tratamientos farmacológicos



Prótesis



Genética



Obstetricia y ginecología



Por dónde empiezo?

Recursos

Nahuel Costa
Inteligencia Artificial y Biomedicina



Andrés Torrubia



Aurelia Bustos

SOFTWARE 2.0

Recursos

Nahuel Costa
Inteligencia Artificial y Biomedicina



Andrés Torrubia



Aurelia Bustos

SOFTWARE 2.0



Jeremy Howard



Rachel Thomas

FASTAI

Recursos

Nahuel Costa
Inteligencia Artificial y Biomedicina



[DotCSV](#)

Divulgador de Inteligencia
Artificial



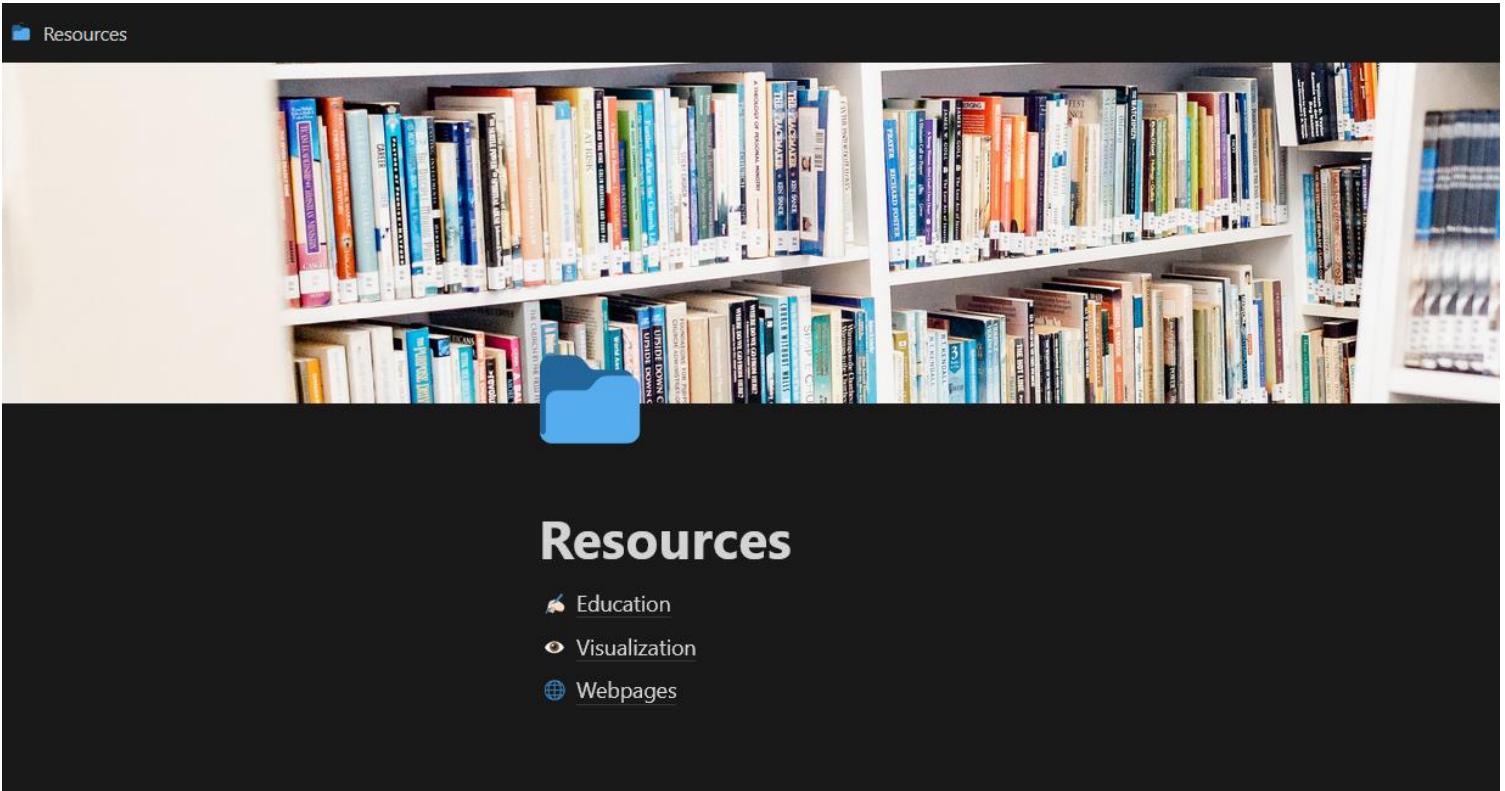
[La hiperactina](#)

Divulgadora biomedicina

Recursos

Nahuel Costa
Inteligencia Artificial y Biomedicina

[Lista de recursos](#)



Perspectivas futuras



Perspectivas futuras

Nahuel Costa
Inteligencia Artificial y Biomedicina

```
● ● ●

import random
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Input, Dense, LSTM, Bidirectional, Masking

# keeping the random seed constant from one experiment to the next makes it
# easier to interpret the effects of hyper-parameters values
seed = 99
random.seed(seed)
tf.random.set_seed(seed)

def create_model(timesteps, input_dim, intermediate_dim, batch_size, latent_dim, epochs,
optimizer):
    timesteps = timesteps
    input_dim = input_dim
    intermediate_dim = intermediate_dim
    batch_size = batch_size
    latent_dim = latent_dim
    epochs = epochs
    if optimizer == 'adam':
        optimizer = keras.optimizers.Adam(learning_rate=0.001)
    else:
        print("unimplemented optimizer")
        exit(-1)
    masking_value = -99.

    class Sampling(keras.layers.Layer):
        """Uses (z_mean, sigma) to sample z, the vector encoding an engine trajectory."""
        def call(self, inputs):
            mu, sigma = inputs
            batch = tf.shape(mu)[0]
            dim = tf.shape(mu)[1]
            epsilon = tf.keras.backend.random_normal(shape=(batch, dim))
            return mu + tf.exp(0.5 * sigma) * epsilon

    # ----- Encoder -----
    inputs = Input(shape=(timesteps, input_dim,), name='encoder_input')
    mask = Masking(mask_value=masking_value)(inputs)

    # LSTM encoding
    h = Bidirectional(LSTM(intermediate_dim))(mask)

    # VAE Z layer
    mu = Dense(latent_dim)(h)
    sigma = Dense(latent_dim)(h)

    z = Sampling()([mu, sigma])

    # Instantiate the encoder model:
    encoder = keras.Model(inputs, [z, mu, sigma], name='encoder')
    print(encoder.summary())
    #

    # ----- Regressor -----
    reg_latent_inputs = Input(shape=(latent_dim,), name='_sampling_reg')
    reg_intermediate = Dense(200, activation='tanh')(reg_latent_inputs)
    reg_outputs = Dense(1, name='reg_output')(reg_intermediate)
    # Instantiate the classifier model:
    regressor = keras.Model(reg_latent_inputs, reg_outputs, name='regressor')
    print(regressor.summary())
    #

    return go(t, seed, [])
}
```

Perspectivas futuras

Nahuel Costa
Inteligencia Artificial y Biomedicina

```
● ● ●

import random
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Input, Dense, LSTM, Bidirectional, Masking

# keeping the random seed constant from one experiment to the next makes it
# easier to interpret the effects of hyper-parameters values
seed = 99
random.seed(seed)
tf.random.set_seed(seed)

def create_model(timesteps, input_dim, intermediate_dim, batch_size, latent_dim, epochs,
optimizer):
    timesteps = timesteps
    input_dim = input_dim
    intermediate_dim = intermediate_dim
    batch_size = batch_size
    latent_dim = latent_dim
    epochs = epochs
    if optimizer == 'adam':
        optimizer = keras.optimizers.Adam(learning_rate=0.001)
    else:
        print("unimplemented optimizer")
        exit(-1)
    masking_value = -99.

    class Sampling(keras.layers.Layer):
        """Uses (z_mean, sigma) to sample z, the vector encoding an engine trajectory."""
        def call(self, inputs):
            mu, sigma = inputs
            batch = tf.shape(mu)[0]
            dim = tf.shape(mu)[1]
            epsilon = tf.keras.backend.random_normal(shape=(batch, dim))
            return mu + tf.exp(0.5 * sigma) * epsilon

    # ----- Encoder -----
    inputs = Input(shape=(timesteps, input_dim,), name='encoder_input')
    mask = Masking(mask_value=masking_value)(inputs)

    # LSTM encoding
    h = Bidirectional(LSTM(intermediate_dim))(mask)

    # VAE Z layer
    mu = Dense(latent_dim)(h)
    sigma = Dense(latent_dim)(h)

    z = Sampling()([mu, sigma])

    # Instantiate the encoder model:
    encoder = keras.Model(inputs, [z, mu, sigma], name='encoder')
    print(encoder.summary())
    #

    # ----- Regressor -----
    reg_latent_inputs = Input(shape=(latent_dim,), name='_sampling_reg')
    reg_intermediate = Dense(200, activation='tanh')(reg_latent_inputs)
    reg_outputs = Dense(1, name='reg_output')(reg_intermediate)
    # Instantiate the classifier model:
    regressor = keras.Model(reg_latent_inputs, reg_outputs, name='regressor')
    print(regressor.summary())
    #

    return go(t, seed, [])

}
```



“Procesa mis datos y aplica una red convolucional para aprender patrones que ayuden a identificar X patología”

Perspectivas futuras

Nahuel Costa
Inteligencia Artificial y Biomedicina

- Creatividad para combinar modelos
- Explicabilidad
- Ética y privacidad

