

Diseño y desarrollo de un sistema de ayuda al  
diagnóstico de enfermedades cardiovasculares a partir de  
electrocardiogramas:  
Aprendizaje profundo semisupervisado

Pablo Martín Redondo

2020



Departamento de  
Tecnología Fotónica  
y Bioingeniería

Universidad Politécnica de Madrid



POLITÉCNICA

# Índice

Contexto

Objetivos

Deep Learning

Procesamiento de datos

Aumento de datos: Redes generativas adversarias

Aumento de datos: Transformaciones

Clasificación de enfermedades

Herramienta de ayuda al diagnóstico

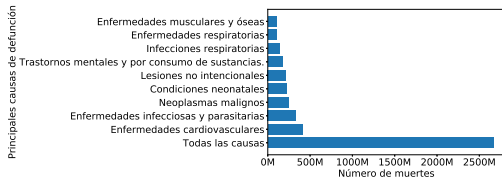
Demo

Conclusiones y líneas futuras

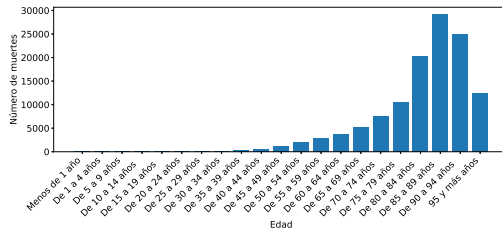
Contexto

# Problemática: Enfermedades cardiovasculares

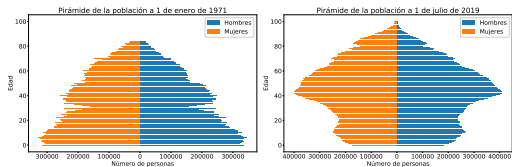
► 413,153 M de defunciones al año



► Riesgo alto en ancianos

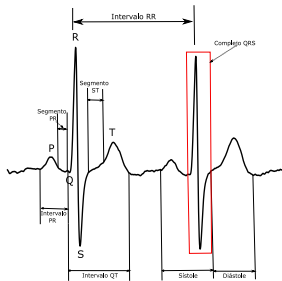


► Población cada vez mas envejecida



# El electrocardiograma

## ► Forma de la señal



## 12 derivaciones

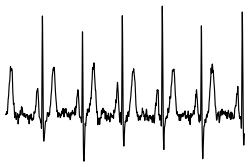
- I
- II
- III
- aVR
- aVL
- aVF
- V1-V6

## ► Procesado clásico: Algoritmo Pam-Tompkins (1985)

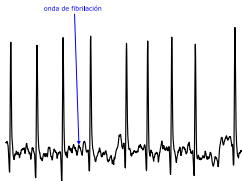


# Enfermedades cardiovasculares

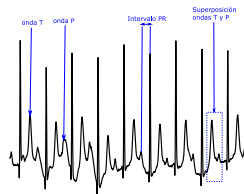
## ► Normal



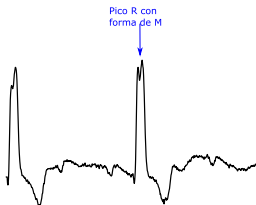
## ► AF



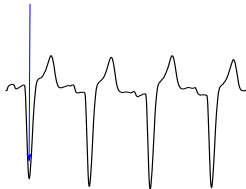
## ► I-AVB



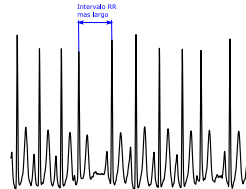
## ► LBBB



## Onda S profunda

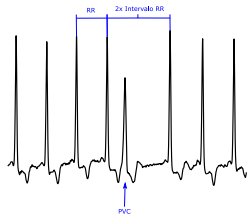


## ► PAC

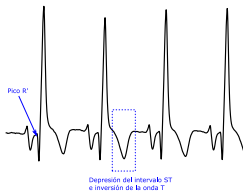


# Enfermedades cardiovasculares

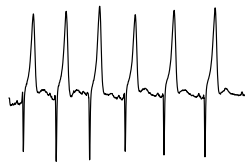
## ► PVC



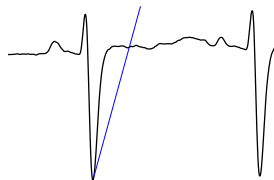
## ► RBBB



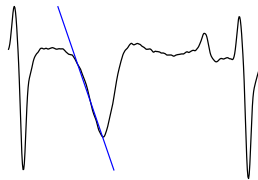
## ► STE



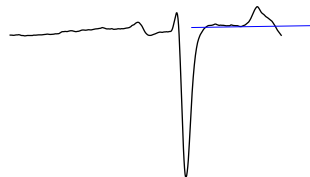
## ► STD



Ascendente



Descendente



Horizontal

## Objetivos



# Diseño y desarrollo de una herramienta de ayuda al diagnóstico

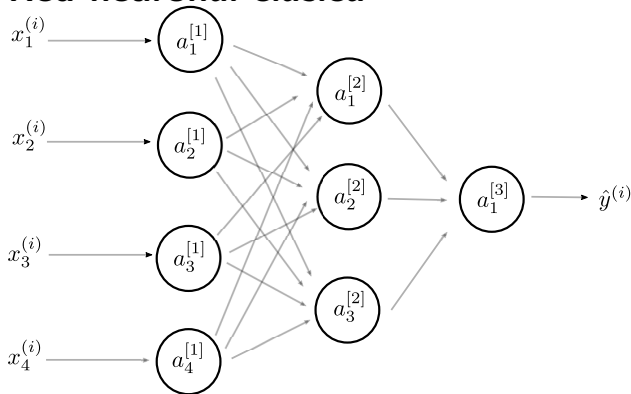
## Objetivos

- ▶ Preprocesado de los datos
- ▶ Datos balanceados
- ▶ Clasificación
- ▶ Herramienta para su uso en entorno clínico

# Deep Learning

# Redes neuronales artificiales

## Red neuronal clásica



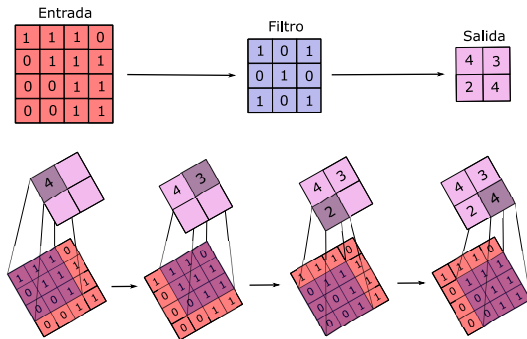
$$A^{[l]} = g^{[l]}(W^{[l]}A^{[l-1]} + b^{[l]})$$
$$A^{[0]} = X, \quad \hat{Y} = A^{[L]}$$

## Elementos

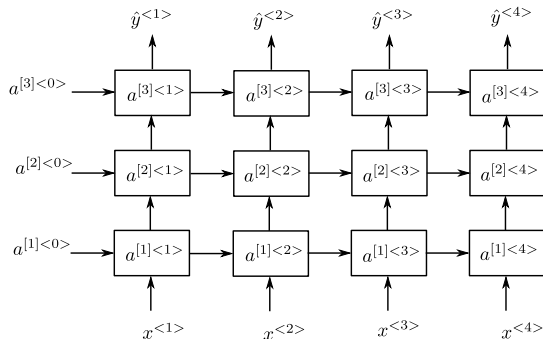
- ▶  $A^{[i]}$  = salida de la capa  $i$
- ▶  $g$  = función de activación
- ▶  $W^{[i]}$  = matriz de pendientes de la capa  $i$
- ▶  $b^{[i]}$  = matriz de ordenadas en el origen de la capa  $i$

# Arquitecturas

## ► Redes neuronales convolucionales



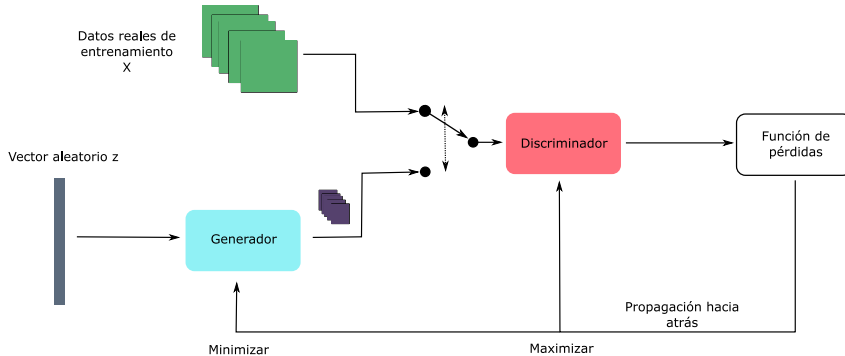
## ► Redes neuronales recurrentes



Simonyan and Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", 2014.

Hochreiter and Schmidhuber, "Long short-term memory", 1997.

## ► Redes generativas adversarias



## Procesamiento de datos

# Los datos de entrenamiento

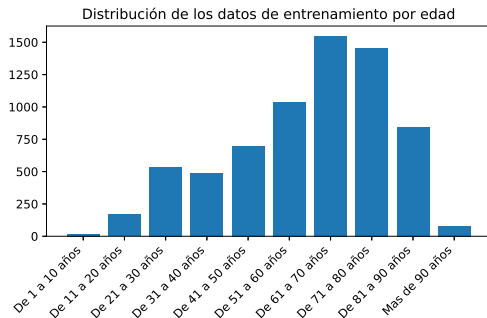
- ▶ Computing in cardiology challenge



## Características

- ▶ 6877 muestras
- ▶ Casi 50/50 por sexo
- ▶ 12 derivaciones
- ▶ 500Hz
- ▶ Entre 6 y 144 segundos

- ▶ Distribución por edad

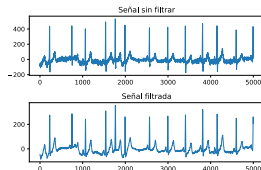
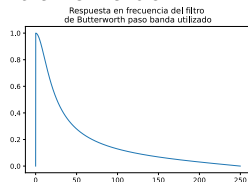


- ▶ [physionetchallenges.github.io/2020](https://physionetchallenges.github.io/2020)
- ▶ Distribución por enfermedad

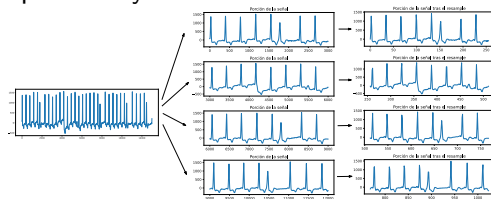
|          | AF    | I-AVB | LB  | Normal | PAC | PVC | RBBB  | STD | STE |
|----------|-------|-------|-----|--------|-----|-----|-------|-----|-----|
| Cantidad | 1.221 | 722   | 236 | 918    | 616 | 700 | 1.857 | 869 | 220 |

# Preprocesado

## 1. Filtrar el ruido



## 2. Separación y submuestreo



## 3. Normalización $\vec{x}_{norm} = \frac{\vec{x} - \min(\vec{x})}{\max(\vec{x}) - \min(\vec{x})}$

### ► Distribución final por enfermedad

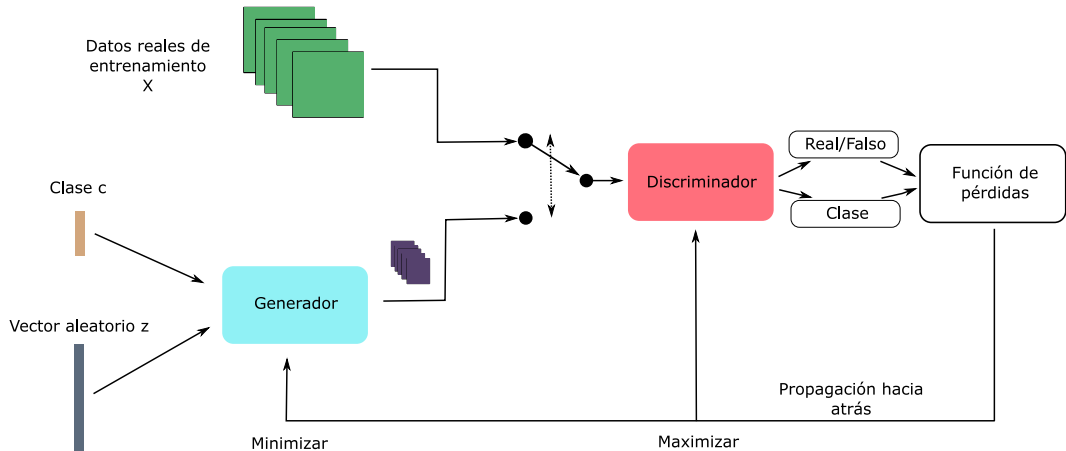
|          | AF     | I-AVB  | LBBB  | Normal | PAC    | PVC    | RBBB   | STD    | STE   |
|----------|--------|--------|-------|--------|--------|--------|--------|--------|-------|
| Cantidad | 27.789 | 15.537 | 5.289 | 21.705 | 19.401 | 24.202 | 41.013 | 21.154 | 5.950 |



Aumento de datos: Redes generativas adversarias

# Arquitectura

## ► Auxiliary Classifier Generative Adversarial Network (AC-GAN)

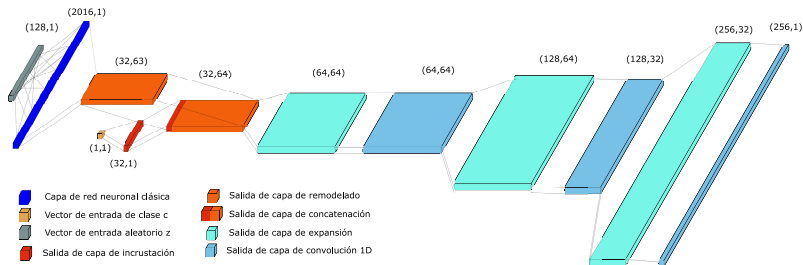


Mirza and Osindero, "Conditional generative adversarial nets", 2014.

Odena, Olah, and Shlens, "Conditional image synthesis with auxiliary classifier gans", 2017.

# Arquitectura del generador

## ► Generador



## Configuración

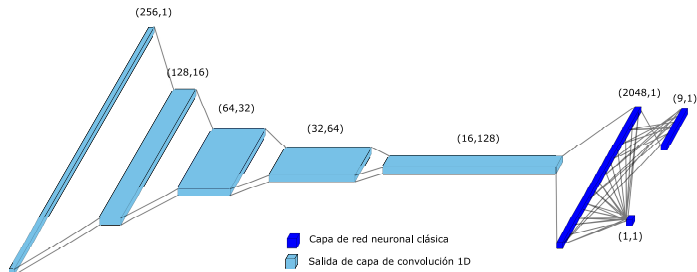
- Recomendaciones DCGAN
- Primera y última capa libres
- Batch Normalization  $\alpha = 0.8$
- Dropout 25%

- *ReLU*
- Salida tanh
- Filtros de 3 unidades
- Padding same

Radford, Metz, and Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", 2015.

# Arquitectura del discriminador

## ► Discriminador



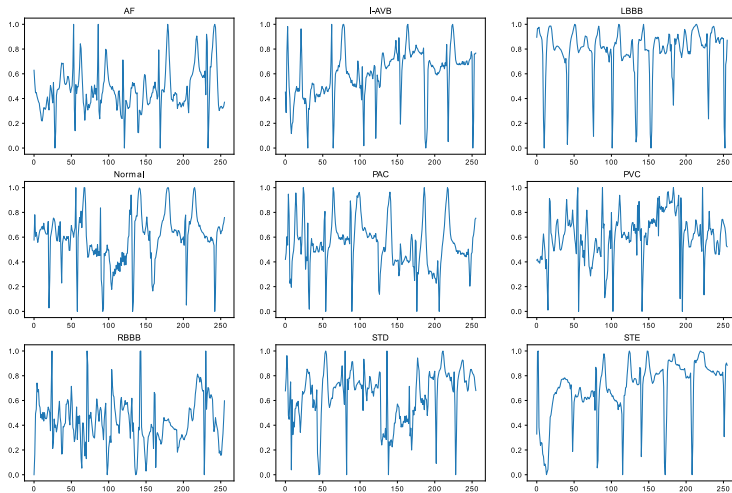
## Configuración

- Recomendaciones DCGAN
- Primera y última capa libres
- Sin batch normalization
- Dropout 25%
- *LeakyReLU*
- Salidas  $\sigma(x)$  y *softmax*
- Filtros de 3 unidades
- Padding same y stride 2

# Entrenamiento y resultados



## ► Muestras generadas

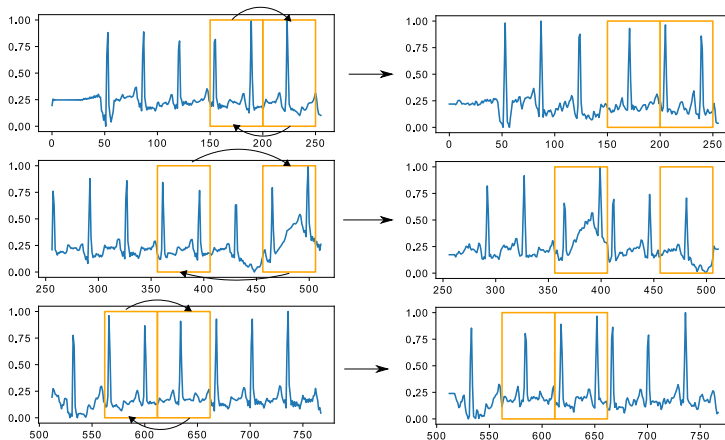


- Adam  $\alpha = 10^{-4}$   
 $\beta_1 = 0,5$  y  
 $\beta_2 = 0,999$
- 200 épocas
- 64 señales por lote
- 4 sesiones
- 12h/sesión
- 280.225 parámetros

Aumento de datos: Transformaciones

# Transformaciones

## ► Proceso de transformación

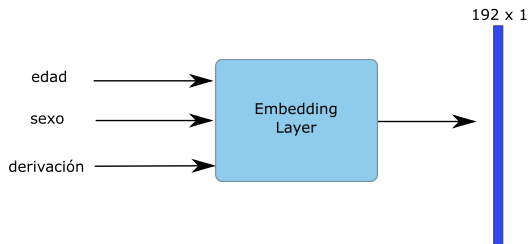


1. Tomar ventanas aleatorias de 50 muestras
2. Intercambiarlas
3. Ruido gaussiano de entre -0.02 y 0.02
4. Filtrado y normalización

## Clasificación de enfermedades



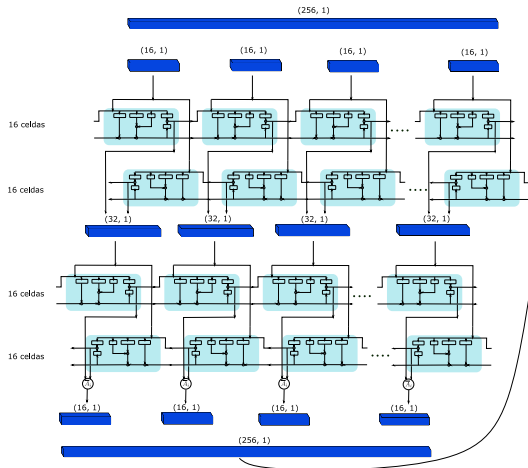
## Arquitectura: primera entrada



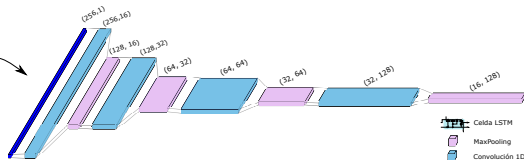
- ▶ Interpretación de datos cualitativos
- ▶ Primera entrada: Embedding3-192
- ▶ Crea una representación de esa entrada en un vocabulario de 192 símbolos

# Arquitectura: segunda entrada

- Interpretación de la señal
- Segunda entrada: 2BiLSTM-Conv4



- Forma + temporalidad

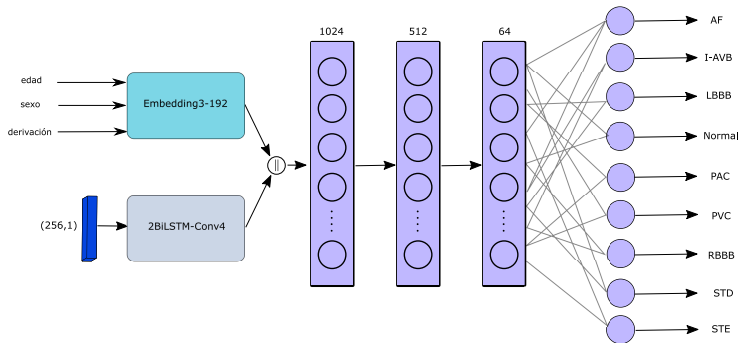


## Capa convolucional

- Filtro con kernel 3, padding same y stride 1
- Batch normalization
- Max pooling  $\frac{1}{2}$
- Dropout

# Arquitectura

## ► Modelo completo

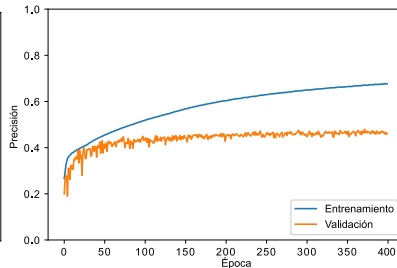
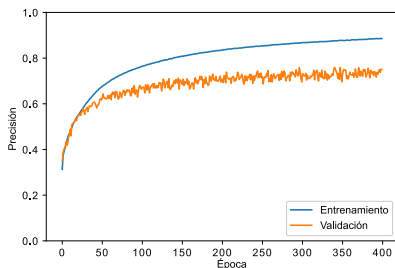


## Hiper-parámetros

- Batch normalization  
 $\alpha = 0,8$
- Dropout 25%
- 4 capas de FCL

# Entrenamiento

## ► Entrenamiento con las 2 técnicas de aumento de datos



## Configuración

- Datos 95/5
- 400 épocas
- 7 horas
- 64 vectores por lote
- Adam  $\alpha = 10^{-4}$ ,  
 $\beta_1 = 0,9$  y  $\beta_2 = 0,999$

# Resultados

## ► Utilizar ambos

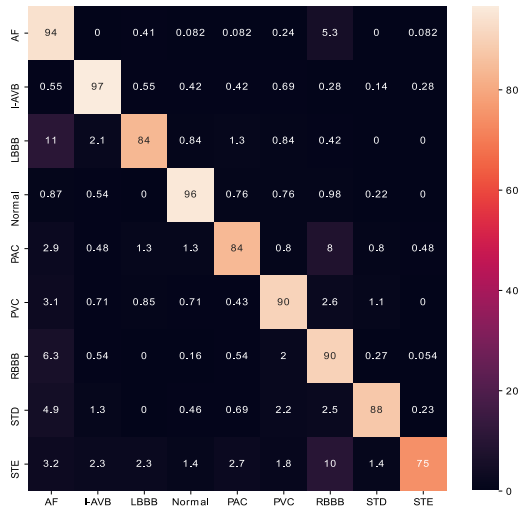
1. Evaluación de los trozos
2. Media aritmética
3. Filtro de sesgos
4. 1 si  $> 0,6$ /máximo

|           | Transformaciones | GANs   | Final  |
|-----------|------------------|--------|--------|
| Precisión | 88,19%           | 94,78% | 95,11% |

► Falsos positivos 0,45%

► Falsos negativos 0,47%

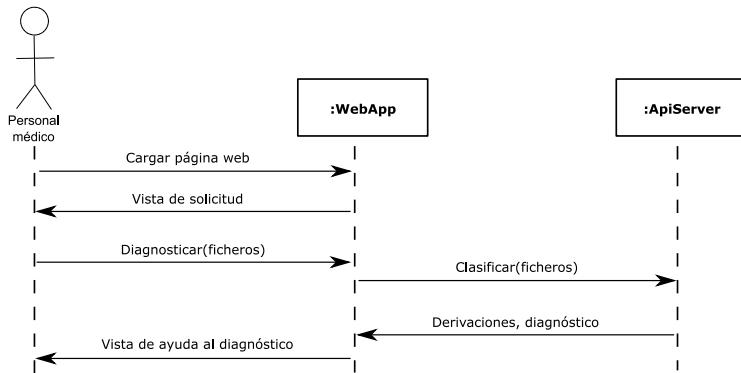
## ► Matriz de confusión final



Herramienta de ayuda al diagnóstico

# Componentes

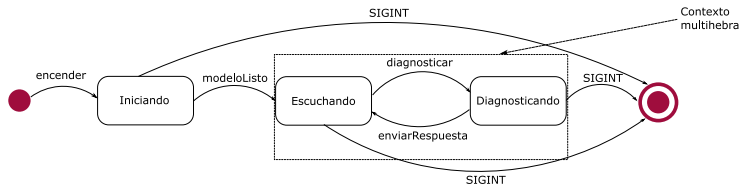
- ▶ Arquitectura orientada a servicios
- ▶ *Como personal médico quiero poder enviar los ficheros necesarios para obtener una ayuda en el diagnóstico de enfermedades cardiovasculares.*



# Componentes

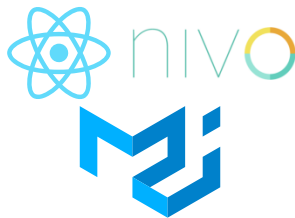
## Proveedor

- ▶ Python
- ▶ Tensorflow
- ▶ Flask
- ▶ Singleton almacén



## Consumidor

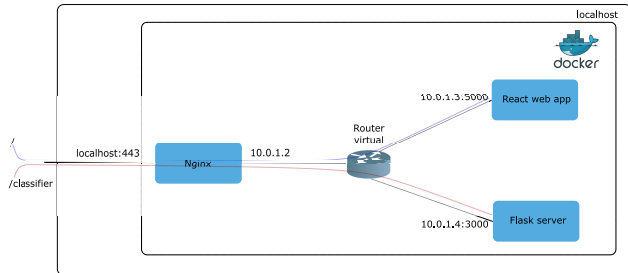
- ▶ React
- ▶ Material-UI
- ▶ nivo





# Despliegue

## ► Despliegue en contenedores



## Contenedores

- NGINX - Alpine Linux
- React - Alpine Linux Node
- Flask - Python container

## Máquina

- CPU 1 núcleo
- 1,75 GB de RAM
- Certificado
- Puertos 80 y 443

Demo

## Conclusiones y líneas futuras

## Conclusiones

### Objetivos

- ▶ Documentación
- ▶ Aumento de datos
- ▶ Clasificación
- ▶ Herramienta
- ▶ Hospital 12 de Octubre

## Líneas futuras

### **Potencial**

- ▶ Continuación
- ▶ Formato de los archivos
- ▶ Gestión de pacientes
- ▶ Mejorar el modelo
- ▶ Utilización entorno real

Gracias