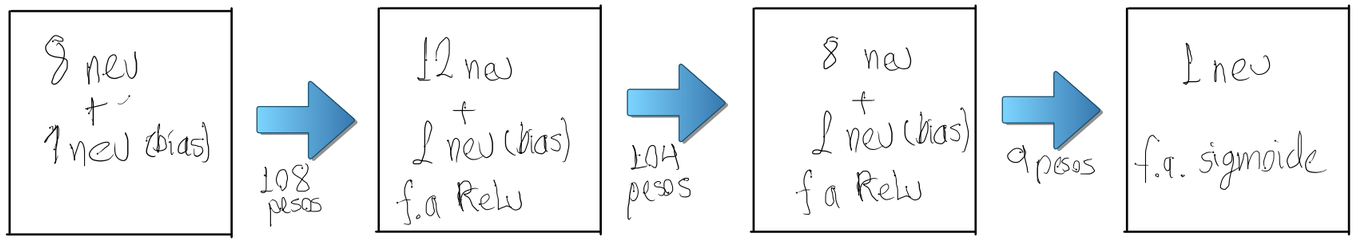
Modelo de Clasificación Binaria

```
modelo = Sequential()
modelo.add(Dense(12,input_dim=8,activation='relu'))
modelo.add(Dense(8,activation='relu'))
modelo.add(Dense(1,activation='sigmoid'))
```




```
modelo.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

$9 \times 12 = 108$

$13 \times 8 = 104$

$9 \times 1 = 9$

221 pesos

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 12)	108
dense_1 (Dense)	(None, 8)	104
dense_2 (Dense)	(None, 1)	9

función de activación en la capa de salida: sigmoide

función de perdida: 'binary_crossentropy'

metrics: accuracy

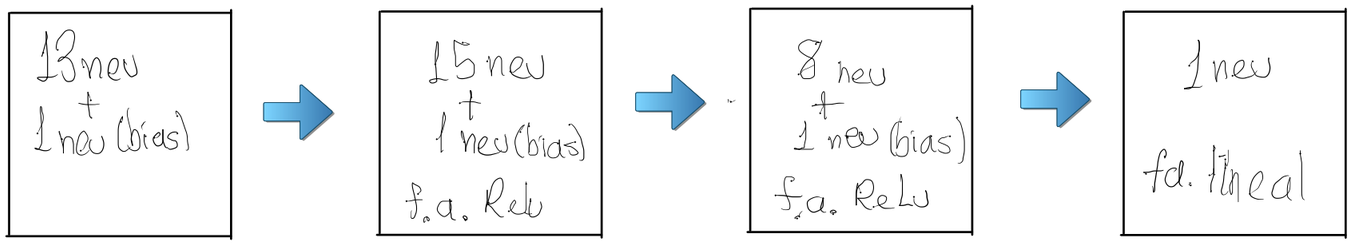
Modelo de Regresión Múltiple

```
modelo = Sequential()
```

```
# Agregamos capas al modelo
modelo.add(Dense(15,input_dim=13,activation='relu'))
modelo.add(Dense(8,activation='relu'))
modelo.add(Dense(1))
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 15)	210
dense_1 (Dense)	(None, 8)	128
dense_2 (Dense)	(None, 1)	9

Total params: 347 (1.36 KB)




```
[ ]  modelo.compile(loss='mean_squared_error',optimizer='adam',
                    metrics=[tf.keras.metrics.RootMeanSquaredError()])
```

función de activación en la capa de salida: $f(x)=x$
función de perdida: 'mean_squared_error'
metrics: RootMeanSquaredError

Modelo de Clasificación Múltiple

```
modelo = Sequential()
```

```
modelo.add(Dense(12,input_dim=4,activation='relu'))
modelo.add(Dense(8,activation='relu'))
modelo.add(Dense(10,activation='relu'))
modelo.add(Dense(3,activation='softmax'))
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 12)	60
dense_1 (Dense)	(None, 8)	104
dense_2 (Dense)	(None, 10)	90
dense_3 (Dense)	(None, 3)	33

Total params: 287 (1.12 KB)


```
modelo.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

función de activación en la capa de salida: softmax
función de perdida: categorical_crossentropy
metrics: accuracy