



CRIPTOGRAFÍA DE CURVAS ELÍPTICAS Y REDES NEURONALES

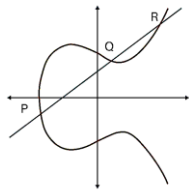
Proyecto final presentado para optar al grado de Ingeniería del Software

Alumno: Javier García Muñoz

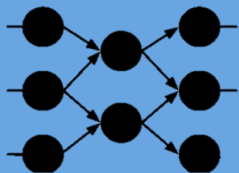
Director: Ángel González Prieto



FUNDAMENTOS MATEMÁTICOS



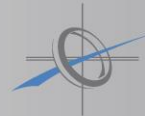
CURVAS ELÍPTICAS




REDES NEURONALES



CASOS PRÁCTICOS

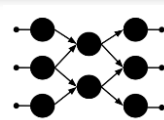



**FUNDAMENTOS
MATEMÁTICOS**



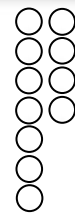

**CURVAS
ELÍPTICAS**





**REDES
NEURONALES**






**CASOS
PRÁCTICOS**



ESTRUCTURAS ALGEBRAICAS


GRUPOS
 $(G, +)$


ANILLOS
 $(R, +, \cdot)$


CUERPOS
 $(K, +, \cdot)$

GRUPOS $(G, +)$

☐ Cerrada

$$+: G \times G \rightarrow G$$

☐ Asociativa

$$g_1 + (g_2 + g_3) = (g_1 + g_2) + g_3$$

☐ Neutro

$$g + e = g$$

☐ Inverso

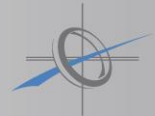
$$g + g^{-1} = e$$

GRUPO
ABELIANO



☐ Distributiva

$$g_1 + g_2 = g_2 + g_1$$



ANILLOS $(R, +, \cdot)$

□ $(R, +)$ es un grupo abeliano y 0 es su elemento neutro

□ La operación binaria \cdot es interna y asociativa en R

$$(r \cdot s) \cdot t = r \cdot (t \cdot s) \quad \forall r, s, t \in R$$

□ Se verifica la propiedad distributiva de \cdot respecto de $+$

$$(r + s) \cdot t = r \cdot t + s \cdot t$$

$$r \cdot (s + t) = r \cdot s + r \cdot t$$

CUERPOS $(K, +, \cdot)$

□ $(K, +)$ es un grupo abeliano


□ (K^*, \cdot) es un grupo abeliano, donde $K^* = K - \{0\}$

□ Se verifica la propiedad distributiva de \cdot respecto de $+$

$$(r + s) \cdot t = r \cdot t + s \cdot t$$

$$r \cdot (s + t) = r \cdot s + r \cdot t$$

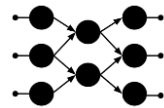
CUERPOS FINITOS F_q
 $q = p^n$


**FUNDAMENTOS
MATEMÁTICOS**



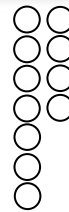

**CURVAS
ELÍPTICAS**




**REDES
NEURONALES**

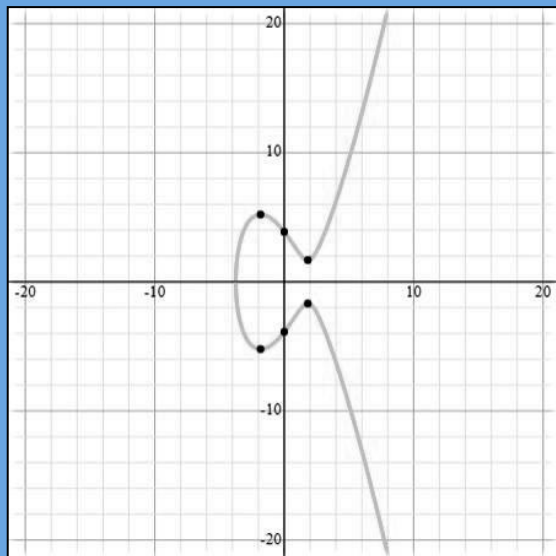



**CASOS
PRÁCTICOS**



CURVAS ELÍPTICAS $E(K)$

Curva algebraica plana de grado 3



Curva elíptica definida sobre \mathbb{R}

$$y^2 = x^3 - 10x + 5$$

- No se cruza sobre sí misma
- No presenta picos

punto en el infinito $O = [0:1:0]$

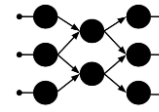
(no tiene correspondencia en el plano afinity)



FUNDAMENTOS
MATEMÁTICOS



CURVAS
ELÍPTICAS



REDES
NEURONALES



CASOS
PRÁCTICOS




ECUACIÓN DE WEIERSTRASS

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

característica del cuerpo es distinta de 2 y 3

$$y^2 = x^3 + ax + b \leftrightarrow a, b \in K$$

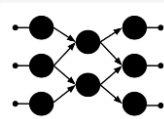
$$4a^3 + 27b^2 \neq 0$$


**FUNDAMENTOS
MATEMÁTICOS**




**CURVAS
ELÍPTICAS**



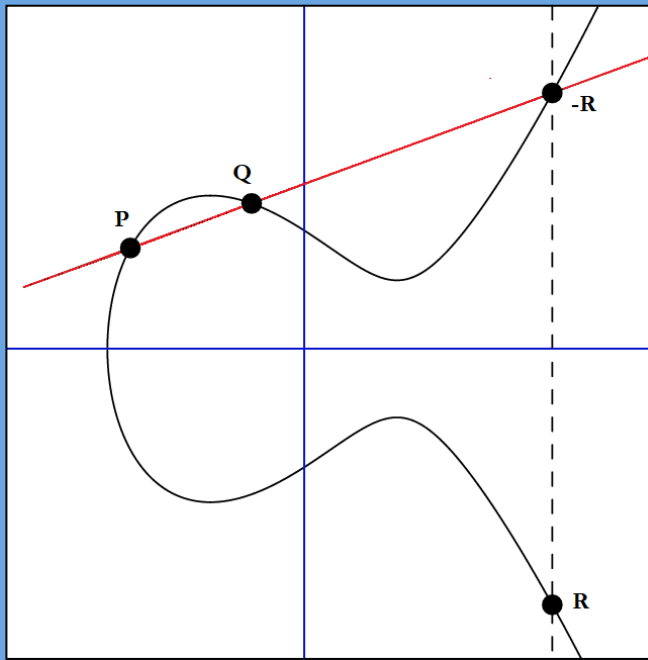

**REDES
NEURONALES**




**CASOS
PRÁCTICOS**



ESTRUCTURA DE GRUPO




método de la cuerda y la tangente

☐ Asociatividad
 $(P + Q) + R = P + (Q + R)$

☐ Neutro
 $P + 0 = P$

☐ Inverso
 $P + P' = 0$

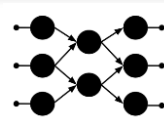
☐ Conmutatividad
 $P + Q = Q + P$


**FUNDAMENTOS
MATEMÁTICOS**



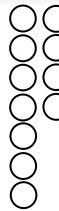

**CURVAS
ELÍPTICAS**




**REDES
NEURONALES**




**CASOS
PRÁCTICOS**

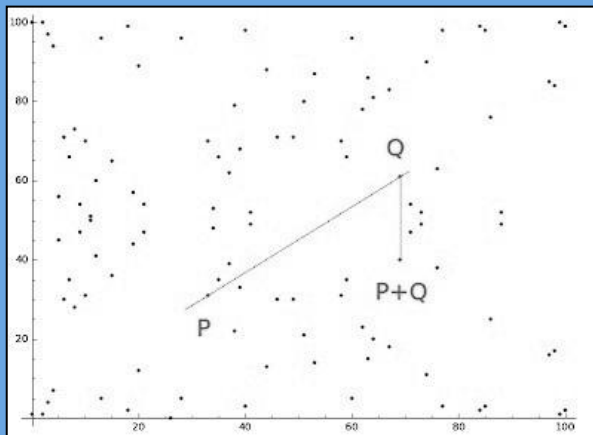


CURVAS ELÍPTICAS SOBRE F_q

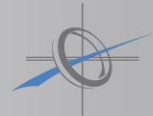
Errores de redondeo
de números reales




CUERPOS FINITOS
(número finito de puntos
cuyas coordenadas
son números enteros)



F_q Potencia de n° primo
 F_{2^m} Potencia de 2 (motivos
computacionales)

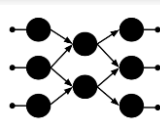



**FUNDAMENTOS
MATEMÁTICOS**



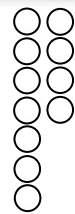

**CURVAS
ELÍPTICAS**




**REDES
NEURONALES**

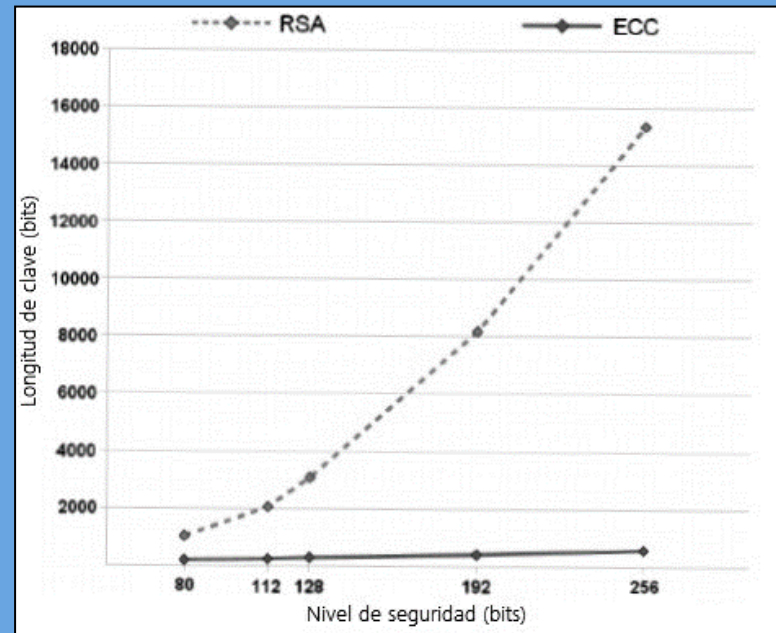


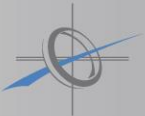

**CASOS
PRÁCTICOS**



USO EN CRIPTOGRAFÍA

N.º bits	RSA	ECC
80	1024	160-223
112	2048	224-255
128	3072	256-383
192	7680	384-511
256	15360	512+

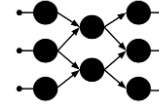




FUNDAMENTOS
MATEMÁTICOS



CURVAS
ELÍPTICAS



REDES
NEURONALES



CASOS
PRÁCTICOS



PROBLEMA DEL LOGARTIMO DISCRETO

DLP

$$x = \log_g(y) \rightarrow g^x = y$$

Exponenciaciones
Productos

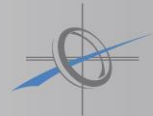
ECDLP


$$Q = n \times P \rightarrow n = ?$$

Productos
Sumas



No hay mecanismos eficientes
para resolver el problema

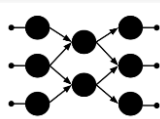



FUNDAMENTOS MATEMÁTICOS



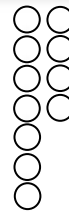

CURVAS ELÍPTICAS




REDES NEURONALES

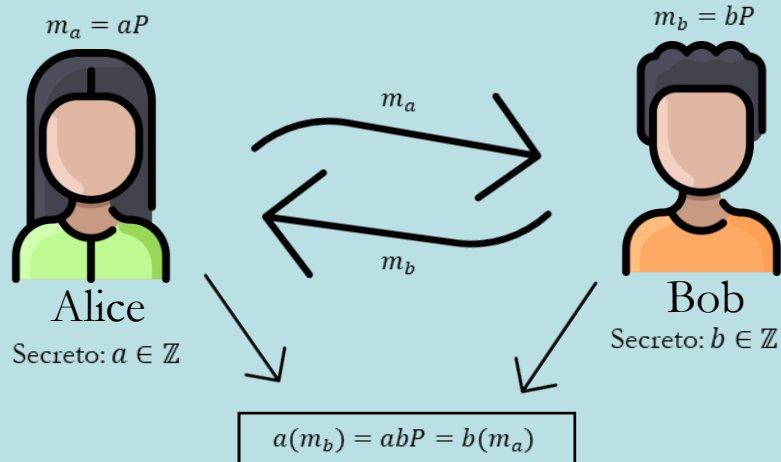



CASOS PRÁCTICOS



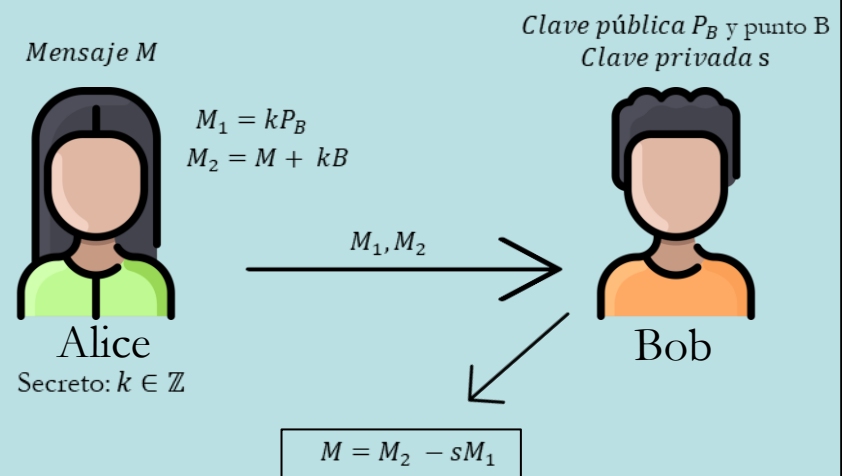
ALGORITMOS ECC

Se acuerda la curva elíptica E sobre un campo finito F_q y el punto $P \in E(F_q)$



Diffie-Hellman

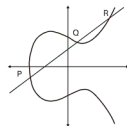
Se acuerda la curva elíptica E sobre un campo finito F_q y el punto $P \in E(F_q)$



ElGamal



FUNDAMENTOS
MATEMÁTICOS



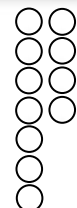
CURVAS
ELÍPTICAS



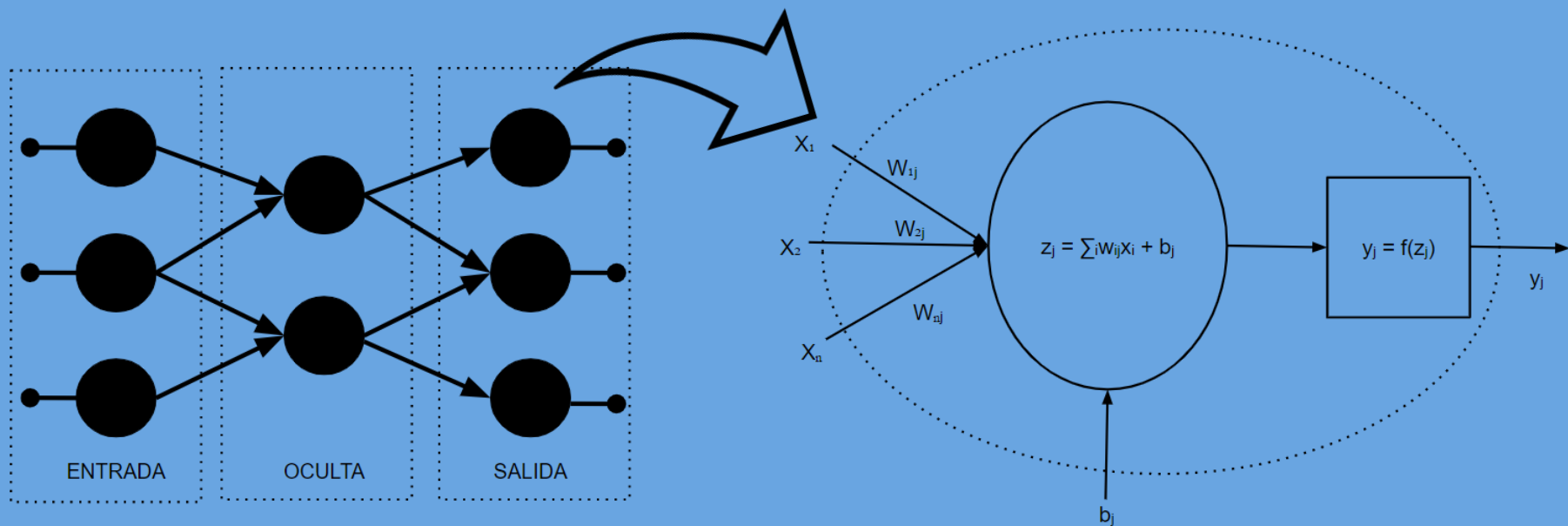
REDES
NEURONALES

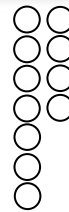
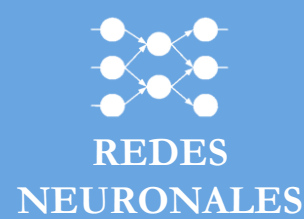
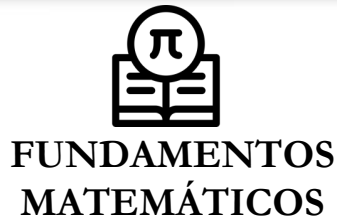


CASOS
PRÁCTICOS



ARQUITECTURA



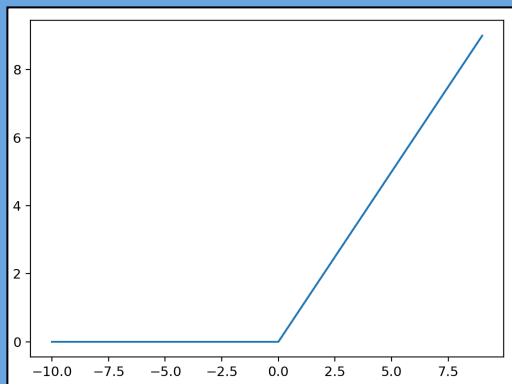


FUNCIÓN DE ACTIVACIÓN

Modifica el valor resultado o impone un límite que se debe sobrepasar para poder proseguir a otra neurona

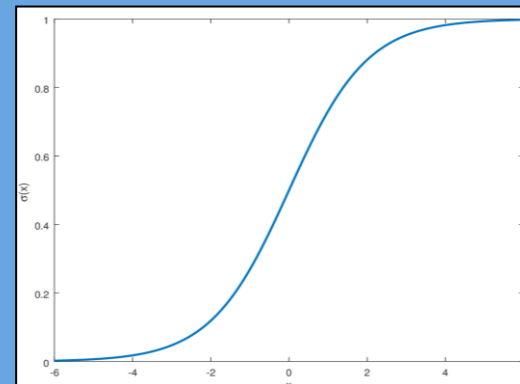
ReLU

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$



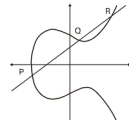
Sigmoide

$$f(x) = \frac{1}{1 + e^{-x}}$$





FUNDAMENTOS
MATEMÁTICOS



CURVAS
ELÍPTICAS



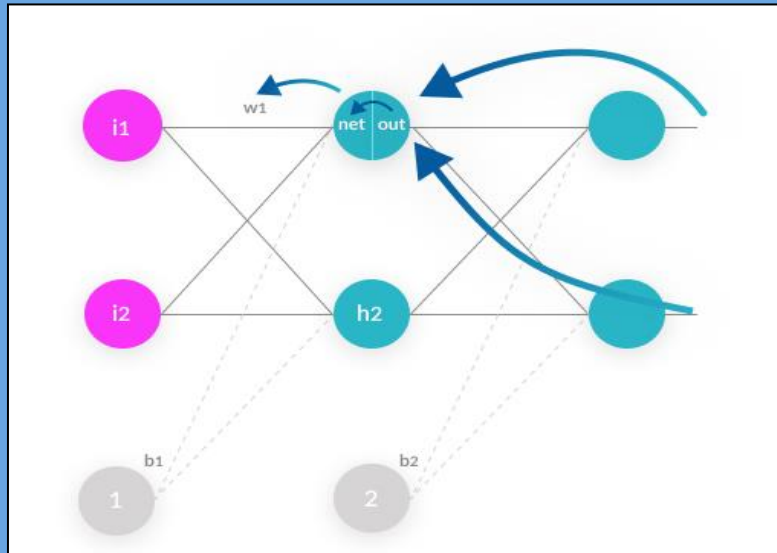
REDES
NEURONALES



CASOS
PRÁCTICOS



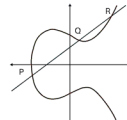
APRENDIZAJE SUPERVISADO



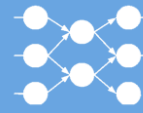
- ☐ Calcular resultado
- ☐ Comparar con la salida correcta
- ☐ Calcular el error
- ☐ Ajustar los pesos



FUNDAMENTOS
MATEMÁTICOS



CURVAS
ELÍPTICAS



REDES
NEURONALES



CASOS
PRÁCTICOS



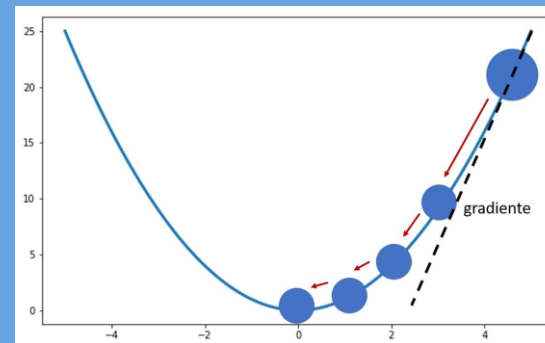
FUNCIÓN DE PÉRDIDA

Evalúa la desviación entre las predicciones y los valores reales.

$$Error = Predicción\ real - Predicción\ realizada$$

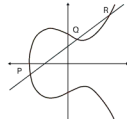
Descenso del gradiente

Algoritmo para reducir el error
(cuidado con desvanecimiento y explosión)

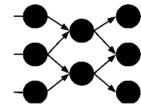




FUNDAMENTOS
MATEMÁTICOS



CURVAS
ELÍPTICAS



REDES
NEURONALES



CASOS
PRÁCTICOS



CASO PRÁCTICO 1

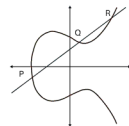
RED NEURONAL CONTRA CIFRADO CÉSAR

CASO PRÁCTICO 2

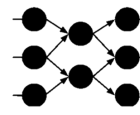
RED NEURONAL CONTRA CRIPTOGRAFÍA
DE CURVAS ELÍPTICAS



FUNDAMENTOS
MATEMÁTICOS



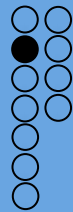
CURVAS
ELÍPTICAS



REDES
NEURONALES



CASOS
PRÁCTICOS



RED NEURONAL CONTRA CIFRADO CÉSAR

DESCRIPCIÓN DEL ALGORITMO

ROT3

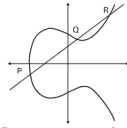
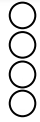
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

EJEMPLO

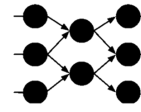
H	O	L	A
↓	↓	↓	↓
K	R	O	D



FUNDAMENTOS MATEMÁTICOS



CURVAS ELÍPTICAS



REDES NEURONALES



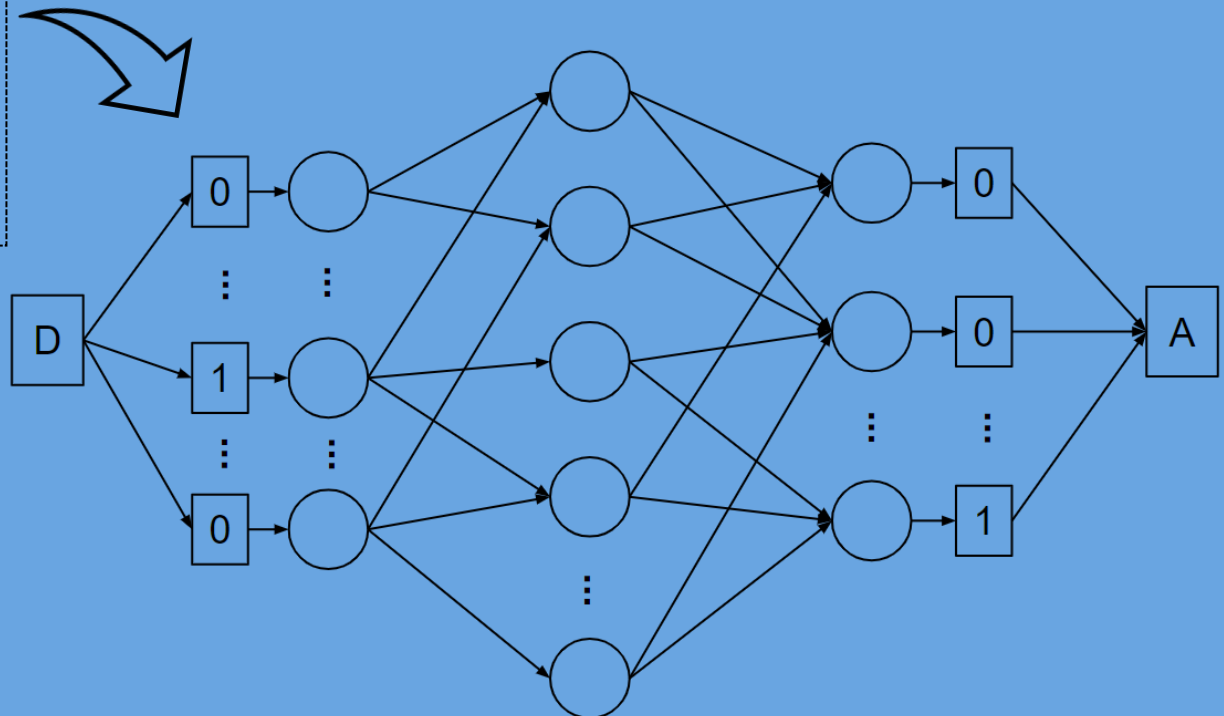
CASOS PRÁCTICOS

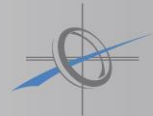


RED NEURONAL CONTRA CIFRADO CÉSAR

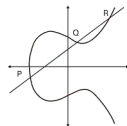
PROPUESTA

A	→	0	→	00000000000000000000000001
B	→	1	→	000000000000000000000000010
C	→	2	→	0000000000000000000000000100
		:		:
Z	→	25	→	1000000000000000000000000000

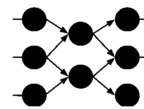




FUNDAMENTOS
MATEMÁTICOS



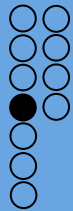
CURVAS
ELÍPTICAS



REDES
NEURONALES



CASOS
PRÁCTICOS



RED NEURONAL CONTRA CIFRADO CÉSAR IMPLEMENTACIÓN



ESTRUCTURA

Generar conjunto de datos

Convertir cadena en bits

Crear modelo


Entrenar modelo

Evaluar resultados

Optimización (GridSearchCV)

```
def generate_words (number_words, test, X_train):
    words = []
    if(test == 0):
        for x in range(number_words):
            words.append(''.join(random.SystemRandom().choice(string.ascii_letters).lower()
                                for _ in range(word_size)))
    else:
        for x in range(number_words):
            repeated = 1
            while(repeated == 1):
                word = ''.join(random.SystemRandom().choice(string.ascii_letters).lower()
                               for _ in range(word_size))
                if(word not in X_train):
                    words.append(word)
                    repeated = 0
            encrypted_words = []
            for word in words:
                cadena = ""
                for x in range(0, word_size):
                    cadena += chr(((ord(word[x]) - 97) + shift) % 26) + 97)
                encrypted_words.append(cadena)
            return encrypted_words, words
```

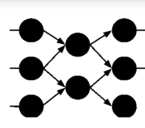



**FUNDAMENTOS
MATEMÁTICOS**




**CURVAS
ELÍPTICAS**




**REDES
NEURONALES**



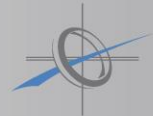

**CASOS
PRÁCTICOS**




RED NEURONAL CONTRA CIFRADO CÉSAR **RESULTADOS OBTENIDOS**

Salida esperada	'efkzrn'	'evymta'	'eufork'	'jhwgml'	'pupwvy'	'anttwe'	'slagap'
Salida devuelta	'efkz-n'	'ivymta'	'eufo-k'	'jhwgml'	'pupwv-'	'anttwe'	'slaiap'

Palabra encriptada = ['mdylhu']
 1/1 [=====] - 0s 22ms/step
 Palabra devuelta por la red ['javier']

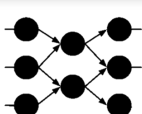



**FUNDAMENTOS
MATEMÁTICOS**




**CURVAS
ELÍPTICAS**



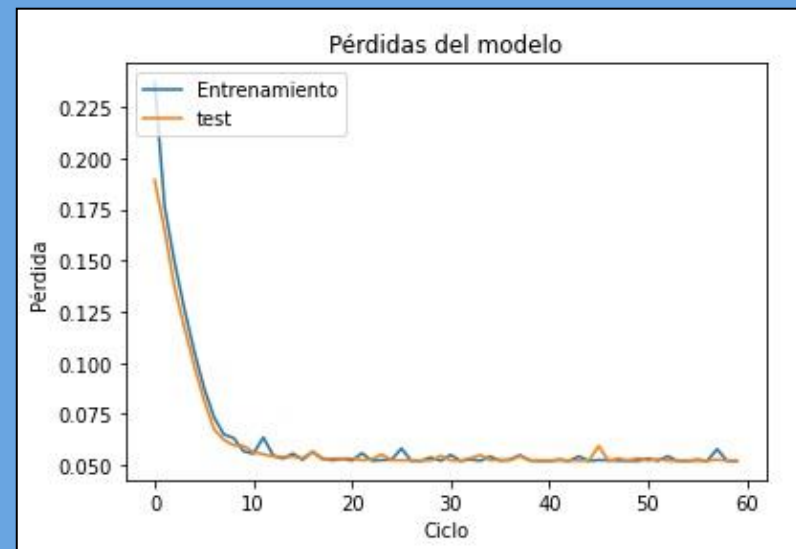
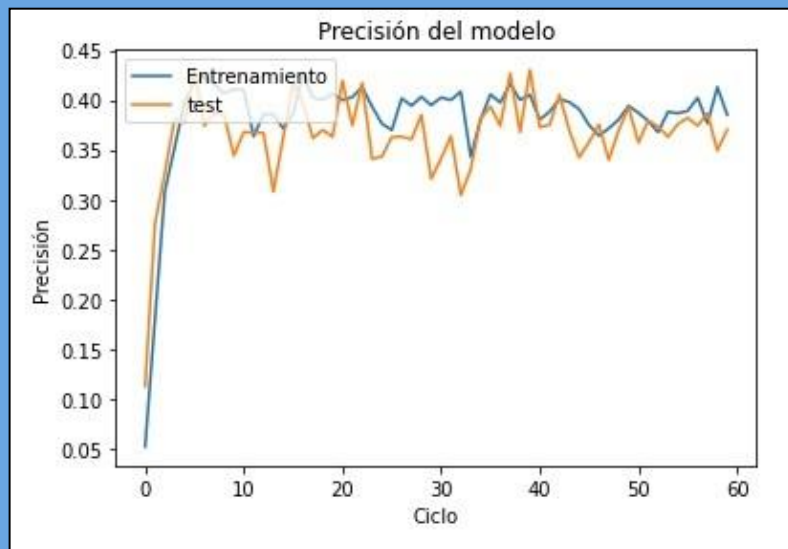

**REDES
NEURONALES**




**CASOS
PRÁCTICOS**

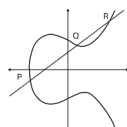


RED NEURONAL CONTRA CIFRADO CÉSAR **RESULTADOS OBTENIDOS**

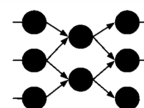




FUNDAMENTOS
MATEMÁTICOS



CURVAS
ELÍPTICAS



REDES
NEURONALES

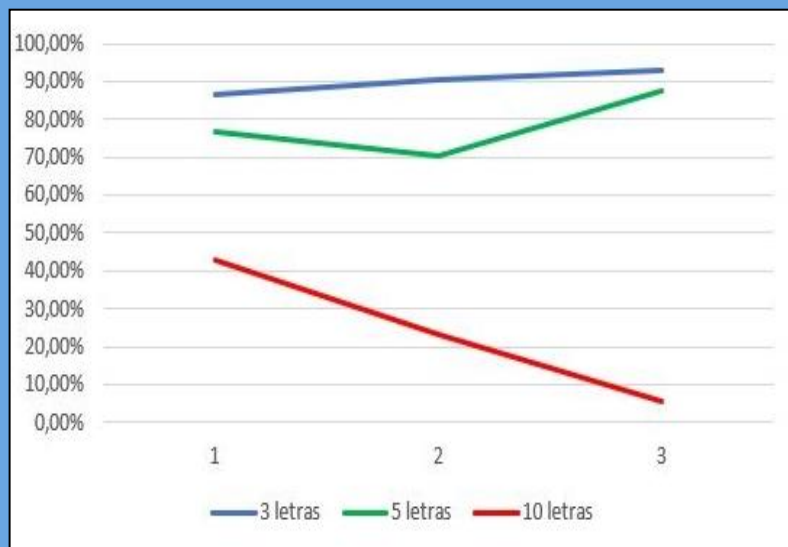


CASOS
PRÁCTICOS

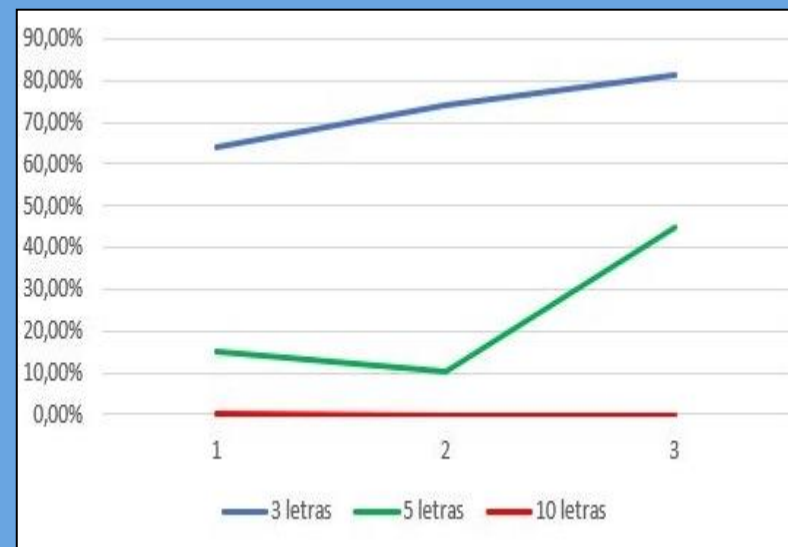


RED NEURONAL CONTRA CIFRADO CÉSAR

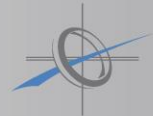
RESULTADOS OBTENIDOS




LETRAS



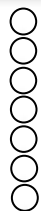
PALABRAS

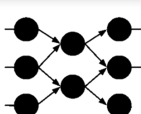



**FUNDAMENTOS
MATEMÁTICOS**



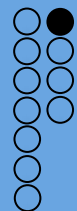

**CURVAS
ELÍPTICAS**




**REDES
NEURONALES**



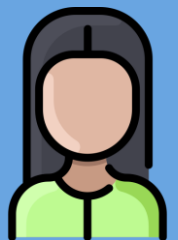

**CASOS
PRÁCTICOS**



RED NEURONAL CONTRA CRIPTOGRAFÍA DE CURVAS ELÍPTICAS DESCRIPCIÓN DEL ALGORITMO ELGAMAL

Se acuerda la curva elíptica E sobre un campo finito F_q y el punto $P \in E(F_q)$

Mensaje M


Alice
 Secreto: $k \in \mathbb{Z}$

$$\begin{aligned}
 M_1 &= kP_B \\
 M_2 &= M + kB
 \end{aligned}$$

Clave pública P_B y punto B
Clave privada s

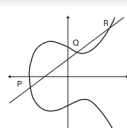

Bob

M_1, M_2

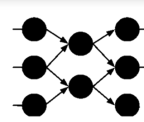
$$M = M_2 - sM_1$$



FUNDAMENTOS
MATEMÁTICOS



CURVAS
ELÍPTICAS



REDES
NEURONALES

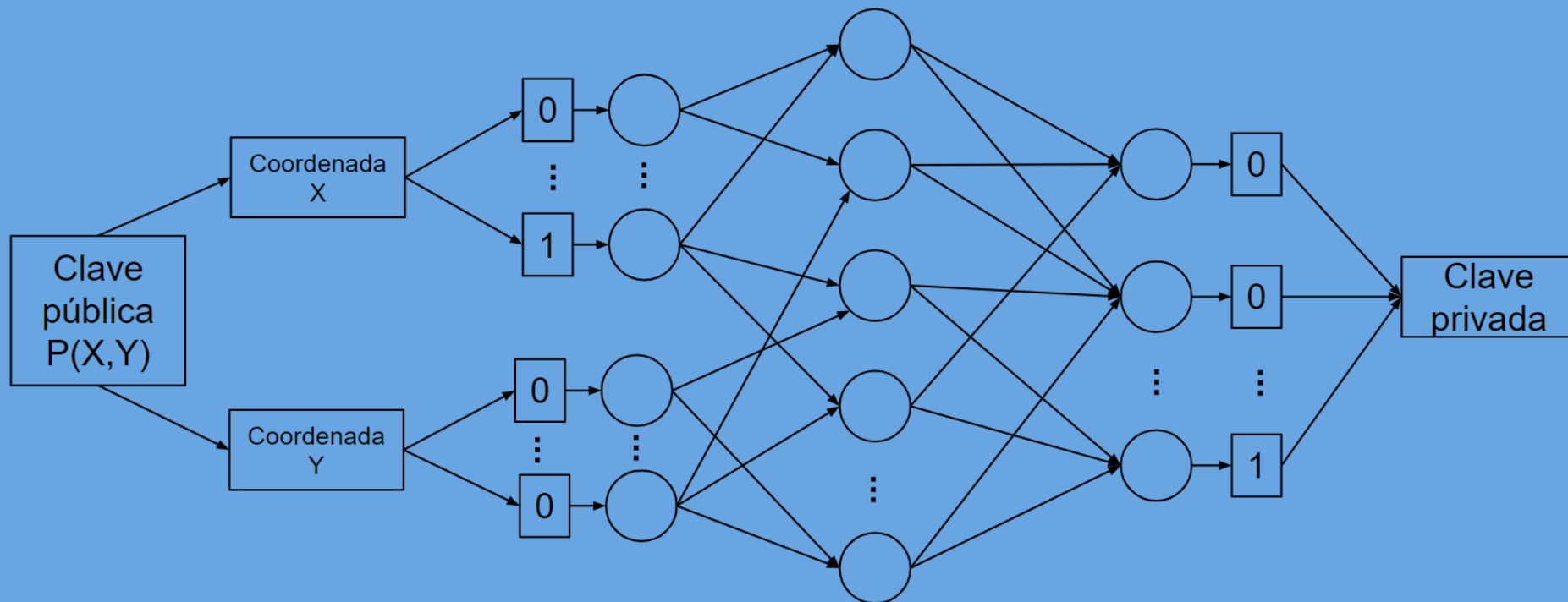


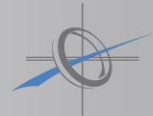
CASOS
PRÁCTICOS




RED NEURONAL CONTRA CRIPTOGRAFÍA DE CURVAS ELÍPTICAS

PROPUESTA



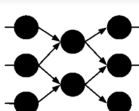



**FUNDAMENTOS
MATEMÁTICOS**




**CURVAS
ELÍPTICAS**




**REDES
NEURONALES**




**CASOS
PRÁCTICOS**



RED NEURONAL CONTRA CRIPTOGRAFÍA DE CURVAS ELÍPTICAS IMPLEMENTACIÓN



ESTRUCTURA

Generar conjunto de datos

Convertir cadena en bits

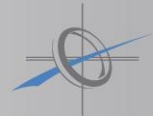
Crear modelo


Entrenar modelo

Evaluar resultados

Optimización (GridSearchCV)

```
def generate_keys (number_keys):  
    pub_keys = np.zeros((number_keys,512))  
    pri_keys = np.zeros((number_keys,256))  
    for x in range(number_keys):  
        pri, pub = gen_keypair(Curve25519)  
        y=255  
        while pub.x != 0:  
            pub_keys[x][y]= pub.x % 2  
            pub.x //= 2  
            y-=1  
        y=511  
        while pub.y != 0:  
            pub_keys[x][y]= pub.y % 2  
            pub.y //= 2  
            y-=1  
        y=255  
        while pri != 0:  
            pri_keys[x][y]= pri % 2  
            pri //= 2  
            y-=1  
    return pub_keys, pri_keys
```

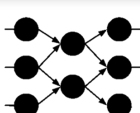



**FUNDAMENTOS
MATEMÁTICOS**




**CURVAS
ELÍPTICAS**




**REDES
NEURONALES**

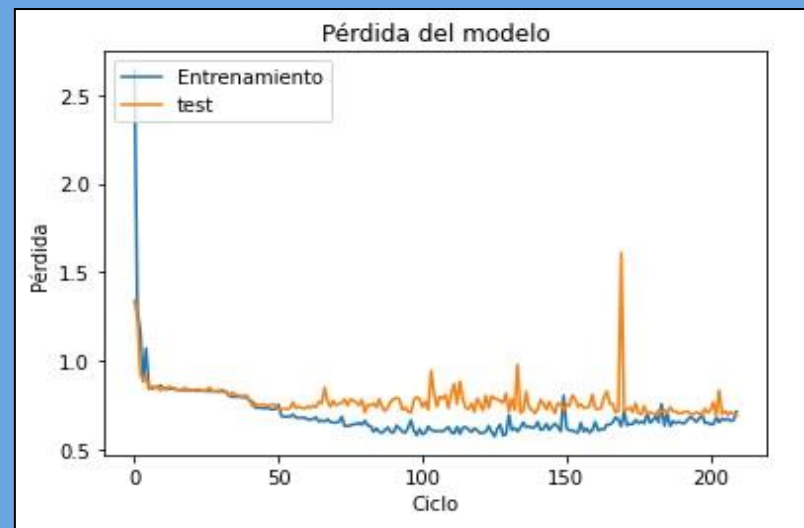
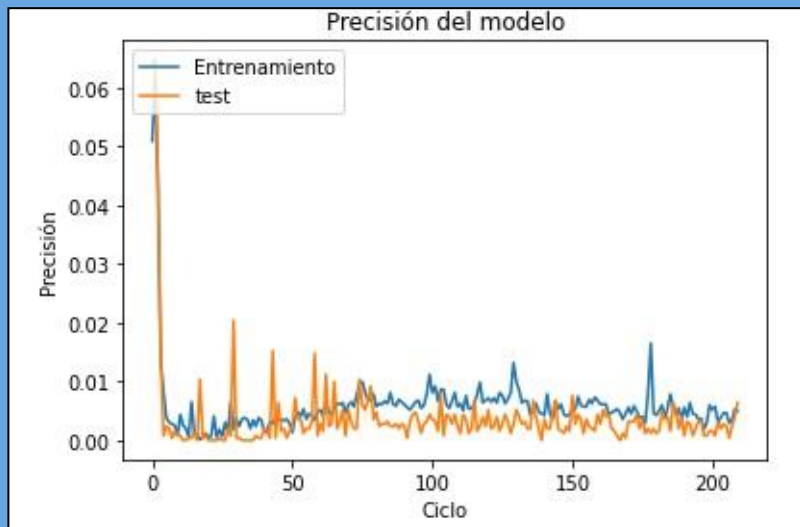



**CASOS
PRÁCTICOS**



RED NEURONAL CONTRA CRIPTOGRAFÍA DE CURVAS ELÍPTICAS





RESULTADOS OBTENIDOS



¿Es PLD resoluble en un tiempo polinomial?



CONCLUSIONES

-  Iniciación en ECC y NN
-  Aprendizaje red neuronal contra CC
-  Aprendizaje red Neuronal contra ECC
-  Mejoras y trabajos para el futuro



**MUCHAS GRACIAS
POR VUESTRA
ATENCIÓN**